user manual

Twentieth edition (covers Impact version 3.48)

Impact user manual

First edition	by John Skingley, October 1995
Second edition	June 1996
Third edition	November 1996
Fourth edition	January 1997
Fifth edition	February 1997
Sixth edition	revised by Hilary Phillips, October 2005
Seventh edition	May 2006
Eighth edition	October 2008
Ninth edition	February 2009
Tenth edition	April 2009
Eleventh edition	December 2009
Twelfth edition	October 2010
Thirteenth edition	April 2011
Fourteenth edition	January 2012
Fifteenth edition	April 2012
Sixteenth edition	May 2012
Seventeenth edition	June 2012
Eighteenth edition	October 2012
Nineteenth edition	April 2014
Twentieth edition	August 2014

Contents

Chapter 1: A guided tour 13

1.1 The iconbar menu 13
1.2 Opening a database 13
1.3 Browsing records 14
1.4 Entering and saving data 14
1.5 Data selection 14
1.6 Indexes 15
1.7 Filters 15
1.8 Field maps 16
1.9 The table display 16
1.10 Making searches 16
1.11 Displaying related data 17
1.12 Making relational links 17
1.13 Using action buttons 17
1.14 Producing reports 18
1.15 Mail merging 19
1.16 Printing labels 19
1.17 More action buttons 20
1.18 Creating a database 20

Chapter 2: An introduction 22

2.1 Program style 22
2.2 Cards 22
2.3 The toolbar 22
2.4 Password protection 22
2.5 Table display mode 23
2.6 Field types 23
2.7 Filters 23
2.8 Indexes 24
2.9 Field maps 24
2.10 Merging data into other programs 24
2.11 Label printing 24
2.12 Import and export 24
2.13 Import and export using drag and drop 25
2.14 Database backup 25
2.15 Disc access 25
2.16 Closing databases 25
2.17 Shortcuts and hot keys 26

Chapter 3: Introduction to tools 27

3.1 The card menu 27
3.2 The toolbar 27
3.3 The browse tools 28
3.4 The data entry tool 28
3.5 The copy tool 28
3.6 The save tool 28
3.7 The close tool 28
3.8 The delete record tool 28
3.9 The select tool 28
3.10 The search tool 28
3.11 The merge tool 28
3.12 The label printing tool 29
3.13 The data import tool 29
3.14 The data export tool 29
3.15 The backup tool 29
3.16 The index, filter and field map tools 29
3.17 The table tool 29
3.18 Finding a record 30
3.19 Finding related data 30
3.20 Editing a record 31
3.21 Closing databases 31
3.22 Copying databases 31

Chapter 4: Filters 33

4.1 Creating and editing filters 33
4.2 Simple filters 33
4.3 Calculation filters 34
4.4 Compound filters 34
4.5 Creating the filter 35
4.6 Deleting a filter 35
4.7 The Hide option 35
4.8 Filter tests 35
4.9 Default settings for filter selection 36

Chapter 5: Indexes 37

5.1 Creating and modifying indexes 37
5.2 Selecting the sort fields 37
5.3 Text fields 37
5.4 The last word 38
5.5 Case matching 38

5.6 The sort order 38
5.7 Using a filter 38
5.8 Manual record selection 38
5.9 Creating the index 38
5.10 Deleting indexes 39
5.11 Default settings for index selection 39

Chapter 6: Field maps 40

6.1 Creating and modifying field maps 40
6.2 Field selection 40
6.3 Concatenating fields 40
6.4 Creating the map 41
6.5 Deleting a field map 41
6.6 Default settings for field map selection 41

Chapter 7: Data entry 42

7.1 Modifying existing data 42
7.2 Using drag and drop 42
7.3 Entering new data 42
7.4 Writable fields 43
7.5 Date fields 43
7.6 File fields 43
7.7 Image fields 44
7.8 Flag and option fields 45
7.9 Menu and browser fields 45
7.10 Saving the data 45
7.11 Clearing fields 45

Chapter 8: Search and replace 46

8.1 The search window 46
8.2 Entering replacement data 47
8.3 Starting the search 47
8.4 Replacing the data 48
8.5 Automatic replacement 48
8.6 Tabulating finds 48

Chapter 9: Tables 49

9.1 The table tool 49
9.2 The table menu 49
9.3 Aligning content of columns 49
9.4 Resizing table columns 50
9.5 Hiding and reordering table columns 50

9.6 Changing the title of the table 50
9.7 Changing the table font and size 50
9.8 Selecting records in a table 50
9.9 Exporting the table as a text file 51
9.10 Exporting as a Draw file 51
9.11 Printing a table 51
9.12 Editing data within the table 53
9.13 Editing several selected records together 54

Chapter 10: Mail merging and formatted reports 55

10.1 Introductory example 55
10.2 The data merge process 57
10.3 The merge window 57
10.4 Selecting a document 57
10.5 Creating a new document 58
10.6 Previewing the document 58
10.7 Permanent merging 58
10.8 Printing reports 59
10.9 Aborting a print run 59
10.10 Sending e-mails 59
10.11 Saving your document 59
10.12 Selecting data records 59
10.13 Entering merge commands from Impact 60
10.14 Technical notes 60
10.15 The Fetch command 61
10.16 The Record command 63
10.17 The Table command 64
10.18 The Scan command 66
10.19 The Calc command 66
10.20 The Date command 67
10.21 Entering merge commands directly from Impression 68
10.22 Entering merge commands directly from Ovation Pro 69
10.23 Entering merge commands directly into ImpEmail 71
10.24 Merging and attachments in ImpEmail 71
10.25 Creating label sheets in Ovation Pro and Impression 73

Chapter 11: Printing labels 75

11.1 The LabPrint application 75
11.2 The label window 75
11.3 Field maps 76
11.4 Printing labels 76

11.5 Saving labels 76
11.6 Queuing labels 77
11.7 Choosing a default label style 77
11.8 Designing labels 77
11.9 The label design window 77
11.10 Selecting a label 78
11.11 Types of label 78
11.12 Selecting a font 78
11.13 Positioning the text 79
11.14 Using images on labels 79
11.15 Rotating text and images 79
11.16 Label sheet design 79
11.17 Saving the design 81
11.18 Using a different sample text in LabPrint 81

Chapter 12: Creating a database 82

12.1 A simple example 82
12.2 Placing and editing labels 82
12.3 Placing and editing a text field 82
12.4 Multi-line text fields 83
12.5 Adjusting the card 83
12.6 Saving the database 83
12.7 Editing a card design 83
12.8 A choice of field types 83
12.9 Multiple cards 84
12.10 Setting a password 84
12.11 Card attributes window 84
12.12 Copying fields 85
12.13 Editing the toolbar 86

Chapter 13: Field types 87

13.1 Why have field types? 87
13.2 Date/time fields 87
13.3 Number fields 87
13.4 Text fields 87
13.5 File fields 87
13.6 Image fields 88
13.7 Flag fields 89
13.8 Option fields 89
13.9 Browser fields 89
13.10 Menu fields 89

13.11 Calculation fields 90
13.12 Sequence fields 90
13.13 Action buttons 90
13.14 Labels 90
13.15 Boxes 90

Chapter 14: Changing field attributes 91

14.1 The Field editing menu 91
14.2 Common features 91
14.3 Editing legends 91
14.4 The field move tool 92
14.5 The field edit box 92
14.6 Field names 92
14.7 Order or caret order 93
14.8 Default entries 93
14.9 Making writable fields non-writable 93
14.10 Justification 93
14.11 Text fields 93
14.12 Number fields 94
14.13 Date/time fields 95
14.14 File and image fields 96
14.15 Flag fields 97
14.16 Option fields 98
14.17 Menu and browser fields 98
14.18 Action buttons 99
14.19 Sequence fields 100
14.20 Box items 100
14.21 Label items 100
14.22 Calculation fields 101

Chapter 15: Relational databases 103

15.1 An explanation 103
15.2 Foreign fields 103
15.3 Creating foreign fields 103
15.4 Foreign field attributes 104
15.5 Type-specific attributes 104
15.6 Entering foreign data 104
15.7 Remote editing 106
15.8 Clearing foreign fields 106
15.9 Opening the record in the remote database 106

Chapter 16: Calculations 107
16.1 Expressions 107
16.2 Syntax definitions 107
16.3 Constants 107
16.4 Control codes in text strings 107
16.5 Reading values from fields 108
16.6 Types of expression 109
16.7 Functions and variables 110
16.8 Table functions 111
16.9 Query functions 111
16.10 Text functions 113
16.11 Date functions 117
16.12 Maths functions 119
16.13 Miscellaneous functions 122
16.14 Conditional calculations 129
16.15 Using flag fields 129
16.16 Using option, menu and browser fields 130
16.17 Using file and image fields 130
16.18 Operators 130

Chapter 17: Action commands 132
17.1 Execution context 132
17.2 Action command syntax 132
17.3 Field assignments 132
17.4 Star commands 134
17.5 Comments 135
17.6 Indexes, filters, and field maps 135
17.7 Record manipulation commands 136
17.8 Document merging and printing 138
17.9 Label printing 139
17.10 Conditional commands 140
17.11 Displaying dialogue boxes 140
17.12 Card commands 141
17.13 Creating variables 142
17.14 Miscellaneous commands 143
17.15 Using action fields as functions 146
17.16 Examples 146

Chapter 18: Exporting data 154
18.1 Simple export using drag and drop 154
18.2 Using the export tool 154

18.3 Selecting fields and records 155
18.4 Choosing an export format 155
18.5 Exporting the data 157
18.6 Formatting the data 157

Chapter 19: Importing data 159
19.1 Import options 159
19.2 Other settings affecting import 160
19.3 Example: importing into an existing database 161
19.4 Example: importing a file to a new database 163
19.5 Further reference 167
19.6 Import and field types 168
19.7 Viewing the results of an import 170

Chapter 20: Caller display 171
20.1 Background 171
20.2 The Caller display unit 171
20.3 Serial ports 171
20.4 Support software 171
20.5 Looking up the number 171
20.6 Taking action 171

Chapter 21: Choices 173
21.1 Record updating 173
21.2 Return key action on cards 173
21.3 On opening a database 174
21.4 Backup 175
21.5 Field menus 175
21.6 Database menu 176
21.7 Table display 176
21.8 Caller display 176
21.9 Placement grid 176
21.10 Default export format 177
21.11 Text selection 177
21.12 Merge document templates 177
21.13 Implementing changes 177

Chapter 22: Making backups 178
22.1 The need for backups 178
22.2 The backup window 178
22.3 Setting the backup location 178
22.4 Making a backup 179

22.5 Automatic backups and warnings 179
22.6 Backup all 179
22.7 Restoring data 180
22.8 Restoring a database which cannot be opened 180

Chapter 23: Deletion 181
23.1 Clearing field contents 181
23.2 Deleting records 181
23.3 Removing fields from a card 181
23.4 Deleting a card 181
23.5 Deleting an entire database 182
23.6 Deleting unwanted Ovation Pro and Impression merge documents 182
23.7 Deleting unwanted action scripts 182
23.8 Deleting unwanted filters, indexes and field maps 182

Chapter 24: Advanced Impact use 183
24.1 System variables 183
24.2 Storing databases in diverse locations 184
24.3 Using Impact over a network 184
24.4 Loading databases by unconventional means 186
24.5 Copying and renaming databases 186
24.6 Changing the order of texts in option, menu and browser fields 187
24.7 Changing the type of a field 188
24.8 Advanced tool editing 189

Appendix 1: Field types 190

Appendix 2: Mouse and key actions in record cards 191

Appendix 3: Date format codes 192

Appendix 4: ImpEmail manual 193

Appendix 5: Obtaining support 201

Index 202

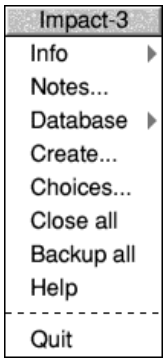
Chapter 1: A guided tour

This section offers an introduction to *Impact* in the form of a hands-on demonstration of the principal features of the program. Full details and explanations of all the features described here are given later in the manual.

To start Impact, double-click on its icon in the directory display, so that it appears on the iconbar.

1.1 The iconbar menu

All Impact features are available from the iconbar menu in the first instance. This is invoked by clicking with the ***Menu*** mouse button over the iconbar icon, in the normal desktop manner.



As you can see, this menu provides access to all databases, via a sub-menu, and provides for the creation of new databases. The database menu can also be obtained by clicking with ***Select*** on the iconbar icon.

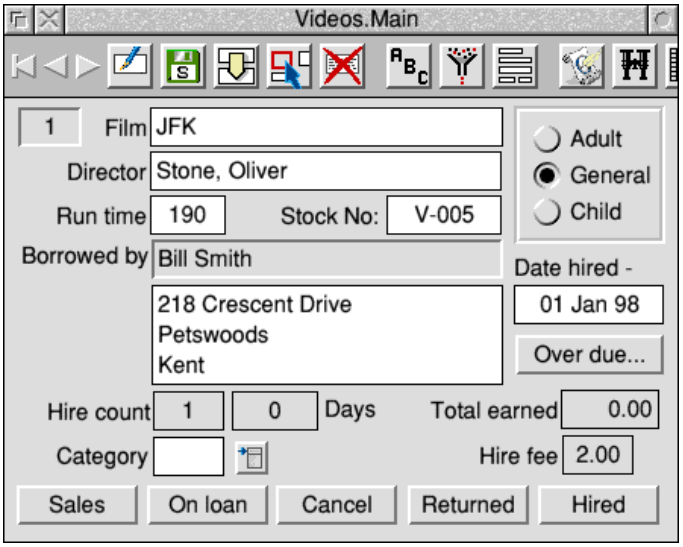
The iconbar menu also offers access to the Impact set-up **choices** window and **backup** facilities.

The *Help* option opens the HTML version of this manual in your web browser (this may not be supplied with the demo version of Impact). Full interactive help is also provided throughout the application, and we strongly recommend that you use this during the demo. Load the RISC OS *!Help* application from your Apps folder, or any of the popular replacements such as *BubbleHelp*, and help messages will appear when you hover over icons and tools with the mouse.

Most of Impact's dialogue windows have a little help icon. If you click on this the relevant chapter of the HTML version of the manual will be opened in your web browser.

1.2 Opening a database

To start, display the database menu, and click on the *Videos* database. When you do this the Videos main **record card** will be displayed. You can have any number of databases open at the same time.



The Videos database represents the type of database which might be used in a video hire shop, giving details of all videos stocked, and the name of the customer, if any, that currently has the video on hire.

This is an example of a **relational database**, where the customer's details are imported from the *Addresses* database, which Impact has opened automatically, although it is not displayed on the screen.

A feature of all database record cards is a **toolbar** along the top, having a set of tool icons. This toolbar will scroll (like the computer iconbar) if there are more tools than can be displayed in the width of the card. The tools displayed on each record card are under the user's control, and custom tools may also be created.

1.3 Browsing records

The arrow icon tools to the left of the toolbar allow you to browse through all the records in the database. The left arrow with a bar may be used to go directly to the first record, while clicking this with the ***Adjust*** mouse button will display the last record in the database.

The other arrow icons conform to normal desktop protocol, and may be used to browse, record by record, through the database. Clicking with ***Adjust*** reverses the browse direction in the normal manner.

1.4 Entering and saving data

The next tool to the right is the **entry tool**. When this is selected, the record card is cleared, and any default entries are inserted. The user may then make further entries or changes that are necessary.

Clicking on the **save tool** will save the data to a new database record, and return the card to **view mode**. However, pressing the ***Return*** key in a writable field will save the data while remaining in **entry mode**, ready for further data to be entered.

Use the pop-up help to explore other tools (select *Help* from the iconbar menu or run the *!Help* application if this is not already turned on). All tools' actions are defined by **action scripts** and are completely configurable by the user.

1.5 Data selection

A database is useless if the data contained cannot be extracted easily. Impact provides three

main ways of selecting data, and these may be used alone or in combination.

Firstly, data records may be ordered using **indexes**. Secondly, particular records may be selected by applying **filters**, to screen out unwanted records. Lastly, the fields in a record may be selected as an ordered set, known as a **field map**.

Each of these selection mechanisms may be defined and named by the user, and selected by pop-up menus at any future time to suit the task in hand.

Indexes, filters and field maps are each defined and selected using the appropriate tools, as described below.

1.6 Indexes

An **index** in Impact provides a means of sorting records into some defined order, either as an aid to finding a particular record, or to facilitate the production of output lists and reports showing records, for example, in alphabetic or date order.

You should now click with ***Menu*** over the index tool and select the *Customers* index from the displayed menu. The tool icon will be shown depressed, indicating that an index in active. The index may be de-activated by clicking again.

With an active index, all database functions will be dependent on the index, including data export, label printing, table displays, etc. If you now browse through the records, they will be displayed in order, sorted on the customer's surname. Note that this is possible although surnames are not stored separately.

Note that only the videos which are currently on hire are being displayed. This is because the index has been created with a named **filter**, so that only a sub-set of records is included. Filters are described below.

As their name suggests, indexes consist of a look-up table, held in memory, giving direct access to the records held on disc. You may create as many as you require, and select them as required. They are automatically kept up-to-date as data is entered and deleted.

Clicking with ***Adjust*** on the index tool will display the index definition window, the use of which is fully described later in the manual.

1.7 Filters

A **filter** in Impact is very much like a filter in real life, in that it lets some records through while keeping others out, as a coffee filter separates the coffee grounds from the liquid. Using filters allows you to extract the required records from a database to suit any situation.

The filter tool is similar to the index tool described above.

If you now click with ***Menu*** on the filter tool, you will be presented with the filter menu, from which you should select the *On Hire* filter. This is the one used in the *Customers* index to select all videos which are currently on hire.

If you deselect the index tool by clicking on it again, browsing records will now display all currently hired videos, but in record number order, since there is no index active. Record order is quite arbitrary, reflecting the order in which data was originally entered, and any subsequent deletions.

As with indexes, any active filter affects all other database operations, such as browsing, export, mail merging, etc. Filters may also be used to filter imported records, via the **CSV import tool** described in *Chapter 19: Importing data*.

Filters are essential components of an Impact database. Unlike indexes described above, they

do not take up a memory by using look-up tables, but are used 'on the fly' to accept or reject records during database operation. Like indexes, you may define as many as you require, and select these from the menu whenever needed.

1.8 Field maps

Having sorted the data records, and filtered out the ones you want, the next thing to do is to decide which individual fields you require from each record. For example, if you wish to print labels you will probably want a name and address, but not a phone number or other information.

Field maps provide the facility to do this, allowing a selection of fields from the card to be chosen in any order. It is also possible to define which fields should be concatenated (joined together with) other fields, and this is useful, for example, to print forenames and surnames on the same line of a label, when they are stored in separate fields. You may also specify characters to be placed between the concatenated fields (usually a single space).

As with indexes and filters, an active field map affects all other database operations where relevant. It will, of course, have no effect on the card display or record browsing, but its full effect may be seen by displaying the database in **table** format.

Before proceeding, click ***Menu*** on the field tool and select the *ReportMap2* field map.

1.9 The table display

If you now click on the **table tool** you will be presented with a tabulated list of all videos which are currently on hire. The table will only show records which pass the selected filter, and the fields defined by the selected field map.

When displaying non-text field types, such as option buttons, user menu selections, etc., Impact will show the text equivalent. For example, if an option button field is displayed, the table will show the text from the selected button.

Despite the many and varied field types available in Impact it is possible to edit most fields in the database directly from the table display by holding the ***Alt*** key and clicking on the cell to be edited. However, if you prefer to edit the data in the record card, just click ***Ctrl-Select*** over the record in the table and that record will be displayed in the record card, allowing instant access to all data editing features.

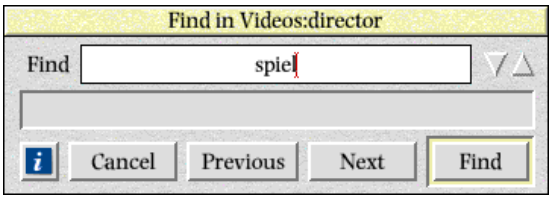
Groups of records may be selected from the table, using ***Select*** and ***Adjust*** in standard desktop manner. This selection may be controlled from a pop-up menu, which also provides other actions, such as deleting and printing.

1.10 Making searches

There are two ways in which a particular record may be found. The **search tool** on the toolbar allows complex search and replace operations and is described in *Chapter 8: Search and replace*.

In addition, a single field search facility, the **find tool**, is provided from the data card. Close the table display, either using the table window Close icon, or by clicking with ***Adjust*** on the table tool. Then turn off any index, filter or field map.

To use the **find tool**, ***Alt-***double-click on the *Director* field on the card. This will cause the Find window to be displayed. This window allows you to type in data to be searched for, and to re-select previously entered data using the arrow icons on the right.



If you now type in 'spiel' followed by the ***Return*** key (or click the *Find* button) the first record containing the director Steven Spielberg will be found and displayed. To find further records, click on the *Next* button. The search will then resume where it left off, and a further film directed by the same director will be found.

Note that the search is case-insensitive, and that it will match any record containing the text supplied. The **find tool** is also available from the card menu. (Click with ***Menu*** over the field to be searched, and choose *Find*.)

1.11 Displaying related data

Now open the databases menu by clicking with ***Select*** on the Impact icon on the iconbar. You will see from this that the *Addresses* database is open, as shown by the ticked menu item. Impact has opened this database in order to display the customers' names and addresses on the Videos card. If you selected the Addresses database from the menu, the addresses data card would be opened.

You can also open the Addresses card from the Videos database. Find a video database record showing a customer (turn on the *Customers* index to find one quickly), and press ***Ctrl*** while clicking with ***Select*** over the *Borrowed by* field. This will cause the customer's record to be displayed in the Addresses database card. This feature is useful where you need access to customer information not shown on the video data card.

1.12 Making relational links

With both record cards on the screen, select a video which has not been hired (turn off any filter or index), and select any customer in the Addresses database. To log the hire of the video to the chosen customer, you might simply drag the customer's name from the Addresses card into the *Borrowed by* field on the Videos card. When you do this, the customer's name and address will be inserted into the card. If you wish to save this change to disc it is then necessary to click on the **Save** tool. Note, however, that other videos have *Date hired* set and *Hire count* increased when hired. **Action buttons** allow us to do all of these things at once.

1.13 Using action buttons

A powerful feature of Impact is its programmable **action buttons**, which greatly speed up and automate many regular operations, such as logging a new video hire.

A row of such buttons have been provided on the bottom of Videos card. If you find another un-hired video, and click on the *Hire* action button (lower-right corner) you will get a message indicating that the hire of a video to the selected customer is about to be logged. If you select OK, the customer's name and address will be inserted into the Videos card, as though you had dragged it there, as described above.

However, several other things have also been done, such as inserting today's date as the *Date hired*, and incrementing the *Hire count* field. Also, the transaction has been saved to disc. Note that you could have designed the database in such a way that all of this could be done manually: programming is not necessary in Impact, it just makes life a lot easier!

To show just how powerful action buttons can be, try the following. First, edit the *Date hired* to

be a few days ago (to simulate the passage of time) and save the change to the record by clicking on the **save** tool (or pressing ***F3***). Now click on the *Return* action button, and click OK on the message window, as before.

Notice that the customer's name has been removed, and the days hired for this video has been increased, as has the total earned amount. Several other things have also been done that are not obvious. If you select the databases menu again, you will see that a database called *Sales* has been opened. Select this database to view its data card.

This database is used to log all sales transactions. If you ***shift-click*** on the **next** browse tool on the Sales card (to move to the last record in the database), you will find the sale you just made has been entered. The record shows the total hire fee, and a VAT breakdown. All this was achieved by the action script executed when you clicked on the *Return* button.

Other action buttons on this card will cancel a hire without logging it as a sale, list all the videos currently on hire, and any videos which are overdue for return, etc. You may view the action scripts associated with these buttons by clicking on the button with ***Adjust***. The commands used are fully described later in the manual, and comments within these scripts will indicate what is going on.

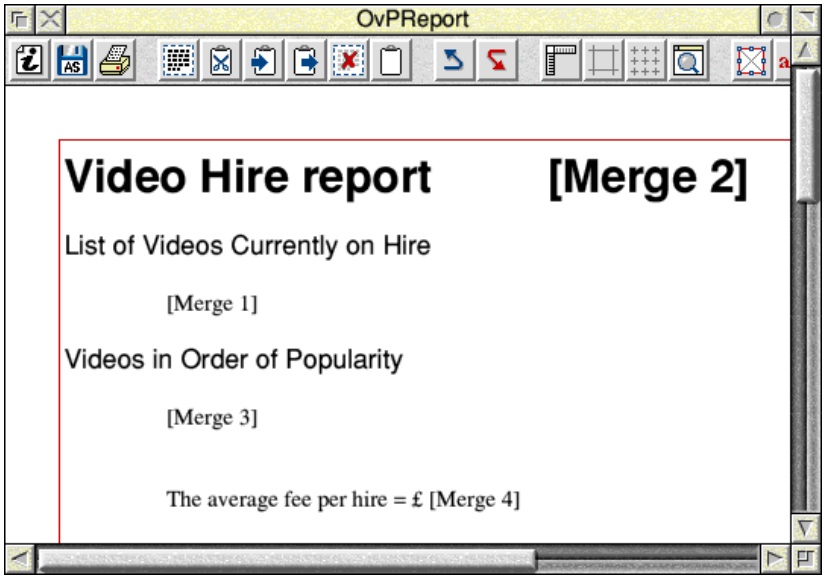
1.14 Producing reports

Impact has the best reporting facility of any database available on RISC OS computers! We can say that without fear of contradiction, since Impact uses both *Impression* and *Ovation Pro* as report writers. These provide all the power of the best desk-top publishing software available, rather than relying on (necessarily) inferior built-in report generators.

Impact does this by interfacing to either of these DTP programs via the *Impulse* module (originally written by Computer Concepts, but now public domain), which allows two-way communication between programs.

To show this in action, click on the **merge tool**. A window similar to the following will be displayed.

If you now click on the *Document* menu icon, and select either *ImpReport* or *OvPReport* (depending on which software you have), the video report document will be loaded into the appropriate application and displayed.


OvPReport
Video Hire report [Merge 2]
List of Videos Currently on Hire
[Merge 1]
Videos in Order of Popularity
[Merge 3]
The average fee per hire = £ [Merge 4]

If this does not happen, it may be that the computer system has not 'seen' either of these programs. In this case locate your copy and install it on the icon bar. Then repeat the report selection as described above.

With the report document on the screen, click the *Test* button on the above window to start the data merge operation. The result should show two tables of information extracted directly from the two databases. Note the current date at the top of the report, the average fee calculation, etc.

The *Print* button will tell the DTP program to merge and print the report, while the *Fix* button will permanently fix the merged data into the document so that it may be edited before printing, if required. If you do this, do not re-save the report so that it overwrites the original un-merged version! The *Clear* button will return the report to its editable state (provided it has not been fixed).

Creating report documents may be done using the *Document* edit button, and this is fully described later in the manual.

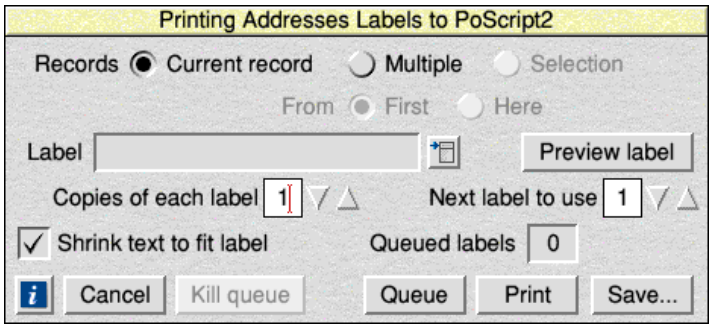
1.15 Mail merging

The **merge tool** can also be used to create single or multiple letters or e-mails, personalised with information on your database. For example: you might wish to e-mail all your contacts to let them know your address is changing, a video hire shop may wish to send messages to recall overdue loans, or a software company may wish to inform registered users about a new version of its product and advise them how much it would cost to upgrade from the version they currently have. Using the *ImpEmail* application such messages may easily be created for electronic communications, while for paper versions *Ovation Pro* and *Impression* can create a document with your customised letterhead, etc.

1.16 Printing labels

Label printing from Impact is handled by a separate application called *LabPrint*, but this action is fully automatic and handled seamlessly from Impact.

To print some labels, go to the *Addresses* database and choose the *AddressLabel* field map from the field map menu. Now select the **Label tool** from the Addresses card. This will display the window shown below.



Click on the *Label* menu icon in this window. The first time you do this LabPrint will be loaded, and its icon will appear on the iconbar. The menu of label types is displayed by LabPrint, and offers a range of pre-defined Avery laser label formats. You can define your own label formats, and this is fully described in the manual.

Select any of the label styles, and its name will be displayed in the Label window. Click on *Preview label* to see how data from the current record would look in this label style.

Choosing the *AddressLabel* field map will ensure that only names, addresses and postcodes are printed. This map also causes a person's name and title to be concatenated and printed on the same line, and ensures that the address fields are in a sensible order.

When first displayed, the Label window is set to print a single copy of the record currently displayed in the data card. If the *Multiple* option is set, all records, or just those from the current one to the end of the database, will be printed. You can also force printing to start on a given label on the sheet, when the sheet is already part used.

To do the printing, click the *Print* button on the Label window, or click the *Save* button to display a standard *Save as* dialogue box, and save the file for printing later. A saved label file may be printed by dragging it onto your printer driver. Label printing is handled by LabPrint automatically.

Alternatively, you can click the *Queue* button to queue individual or groups of labels for printing later when convenient. When there are labels queued, the *Print* and *Save* buttons will operate on the queue.

1.17 More action buttons

An action button *Q-Label* on the Addresses data card automatically selects the appropriate field map, and label style, and queues a label, without the user having to do all of these singly.

The *Write to* button will load a letter 'template' file, merge the current name and address permanently, move the merged document into the Letters database, and log the event. A similar button *Send* will create a fax, and merge the fax number for automatic transmission via *ArcFax*. The *Letters* button will open the *Letters* database.

A phone call may be dialled via a modem by clicking the *Call* button next to the phone number, and an e-mail may be created by clicking the *Mail* button next to the e-mail field. Suitable internet software must be available for this to work. (This could also give direct access to web sites, etc.)

1.18 Creating a database

You may be thinking, at the end of this, that setting up a database must be a complicated operation. Not so! If you select *Create* from the iconbar menu, you will be presented with a blank card, and a selection of fields and other items in a separate window. All that is required is to drag the required fields onto the blank card, give the database a name where indicated, and click the *Save* button. Done!

Well, OK, there's a lot more that can be done. Such as naming fields, setting various options, default entries, date formats, etc. These things are simply achieved by selecting from the card menu. Tools may be dragged onto the toolbar, and re-arranged as required by further dragging. And all this may also be done later, when the database is in use. (Try *Edit card* from the card menu, while browsing one of the sample databases.)

Chapter 2: An introduction

This chapter briefly describes the main facilities offered by Impact and introduces some general concepts and terminology. Further details of all facilities described may be found in the following chapters.

Impact is a general purpose relational database program, providing facilities for the creation, editing, import and export of data, as well as allowing data to be merged into *Impression*, *Ovation Pro, ImpEmail* or other *Impulse* supporting programs, for direct mail shots, report generation, etc.

The data contained in any database may be pure text, numbers, dates, flags indicating on/off or yes/no conditions, scanned images such as personnel photographs, user defined pop-up menu selectors, calculated fields, etc.

Impact also allows programmed action buttons to be included on a data record card, to carry out often repeated actions with a single click.

2.1 Program style

While you will find sufficient information in these pages to allow you to use this program, if you are not wholly familiar with normal RISC OS desktop operation, you are advised to study the computer *User Guide* supplied with your computer. Throughout this manual you will be expected to understand the terms ***click***, ***drag***, etc., and to know the mouse buttons ***Select***, ***Menu***, and ***Adjust***. These terms and action are standard across all conforming RISC OS programs, so if you are unsure of these terms, please consult the *User Guide* before reading on.

2.2 Cards

Impact databases are designed simply by designing a **record card**. This is similar to a card in a card index box, and may be designed by selecting and positioning individual data fields in any desirable manner. They may also display calculation fields, showing the results of calculations made using data from other fields. Calculations are not stored, but are computed whenever the card is displayed or updated. Cards may also contain items which are repeated on all records, such as text legends, etc.

Several cards may be designed for one database, and the fields repeated on several cards, or shared between them. These may be used for specific purposes where the full data available is not required, or where it may be desirable to hide sensitive data.

2.3 The toolbar

Each record card has its own toolbar displaying a row of tools giving instant access to all the main facilities, from simple record browsing to report generation and label printing.

When a new card is created, a default set of tools will be placed on the toolbar. Tools can be added, removed or arranged in a different order using the card editing facilities. Where a database has several cards designed, some of them may not need any tools, in which case a toolbar need not be added.

See *Chapter 3: Introduction to tools* for a description of each tool and *Chapter 12: Creating a database* for details on how to design toolbars for your own databases.

2.4 Password protection

There are situations where people may require access to a database who should not be allowed to see or alter certain sensitive information. To allow for this, it is possible to set password protection on a card. Once this has been done, the card may not be displayed without the user

entering the required password. See *12.10 Setting a password*.

2.5 Table display mode

In addition to the card display which shows the content of one data record at a time, a table display is also provided which shows many records in a scrollable window similar to a spreadsheet display.

Use ***Ctrl-Select*** over any record line in this display to open that record in the card. Most types of data can be edited direct in the table by holding the ***Alt*** key and clicking in the relevant cell. The table display may also be used to define a database record selection as described later.

A table menu is also available to control record selection and multiple record deletion. Tables can be exported as text, as Draw files or printed.

See *Chapter 9: Tables*.

2.6 Field types

Each data field may hold one of several different data types, and these are presented using familiar RISC OS desktop features, such as radio buttons, pop-up menus, etc.

The most common of these is the text type, used to hold all manner of textual data, such as names and addresses, and these fields may be single or multi-line.

While it is possible to hold any data such as a number or date as text, this method has several disadvantages. Such numbers cannot be used in calculations, they cannot be used to sort records into order (200 would count as higher than 1000 as text, since 2 is higher alphabetically than 1), and large numbers would take up more disc space.

For these reasons special field types are defined for non-text fields such as numbers and dates, and these should be used whenever appropriate.

Special fields are also provided to hold images, flag or on/off indicators, multiple option buttons, user defined pop-up menus, and files.

A file field enables any computer file or directory to be linked to a record. Clicking on such a field is equivalent to double clicking on the file in a filer window, causing the file to be run, loaded into a text editor or a DTP program, etc. It could also, depending on the file type, run an Acorn Replay movie, a music file, etc.

See *Chapter 13: Field types*.

2.7 Filters

A filter in Impact, as in real life, is a device which will only let certain things through, holding others back. What passes through is what we want, what remains is discarded, as with say a coffee filter. You may define many different named filters for the same database, and select which one you want to use, so that, for example, only certain people will receive a mail shot, and nobody else.

For complex situations, it is possible to combine two or more filters. For example, we may want the addresses of all club members over 18 who have not yet paid their subscriptions. Filters may be combined in many ways, and saved as **compound filters** for use on future occasions. Compound filters may themselves be used in other compound filters, to construct any desired filter criteria.

See *Chapter 4: Filters*.

2.8 Indexes

In a card index box it is normal to store cards sorted in some way for easy reference. In Impact records are stored in whatever order they are entered, with new records re-using the space vacated by any previously deleted record. This is because moving data about on the disc is not only time consuming, but adds to disc wear and tear.

Instead an index is kept, like that found at the back of a book, allowing any required record to be quickly found. This index lists records in ascending or descending order of a specified field or group of fields. This method allows several indexes to be kept, allowing the appropriate one to be selected for the current purpose. The actual order of records on the disc is not affected.

Impact allows you to build named indexes, on as many fields as you like, sorting in ascending or descending order. It is even possible to create an index which only contains certain selected records, and this will be found to be useful when exporting data, or producing reports and mail shots. This is done using filters.

See *Chapter 5: Indexes*.

2.9 Field maps

It is often necessary to select the data fields to be used in an operation, so that for example a label may include a name and address but exclude other data. Impact provides the facility to create named field selections, defining both the fields and their order.

As with indexes and filters, these named field selections, or maps, may be saved for future use.

See*Chapter 6: Field maps*.

2.10 Merging data into other programs

An important feature of Impact is its support for the *Impulse* module. This public domain module allows all programs written appropriately to communicate with each other, both requesting and sending data, or sending and receiving commands. Using this system, Impact is capable of sending data to *Ovation Pro, Impression* or *ImpEmail*, when asked, and also instructing these applications to merge data, and print/e-mail the results.

See *Chapter 10: Mail merging and formatted reports*.

2.11 Label printing

This is such an important feature that a separate application has been developed to provide it, called *LabPrint*. However, this is controlled directly from within Impact in a seamless manner.

Labels may be dragged directly to the printer, or saved to disc and printed later, perhaps many times. It is also possible to queue labels, singly or in groups, maybe from different databases, and then print them in one go when convenient.

The LabPrint application has a large range of Avery labels pre-defined, and also allows other styles, shapes and sizes to be created and saved.

See *Chapter 11: Printing labels*.

2.12 Import and export

Impact is also fully equipped to import and export data, to and from other applications or disc files, in the conventional desktop manner. Direct in-memory transfer will be used where this is supported by the other application, otherwise transfer will be via a scrap file on disc.

It is likely that you may already have important information stored within another application, such as a spreadsheet or another database package, and may wish to import this data directly

into an Impact database. This may be done, provided the other application can export its data in comma- or tab-separated value, i.e. CSV or TSV format, which most do.

Impact has been made to import and export data in CSV standard, with provision for handling some other variants. It is also possible to export records one field to a line, which is handy for printing a single label, or exporting an address directly into *!Edit*, or other text editor, etc.

See *Chapter 18: Exporting data*, and *Chapter 19: Importing data*.

2.13 Import and export using drag and drop

Text or images may be dragged into an Impact record card from another application. For example, a name or address could be selected in an e-mail or wordprocessing document, and imported directly into a field on the data card. See *Chapter 7: Data entry*.

In the same way, single fields or a selection of fields may be dragged from an Impact record card and dropped into many other applications. If the receiving application supports the RISC OS drag and drop protocol, a ghost caret will appear in the document for you to position where you wish to drop the data. It is also possible to save data to a Filer window using drag and drop. Which fields get exported when dragging is affected by the choice of field map and the selection tool. See *Chapter 18: Exporting data* for more details.

2.14 Database backup

It is very important with any computer application that a backup copy of stored data should be made at regular intervals to guard against loss of valuable information. Impact provides a backup monitoring system which ensures that you cannot quit the program without being warned that a backup is required. Provision is also made to simplify the saving and restoration of backup copies of all databases.

Impact will keep a selectable number of past backup copies of each database, and allow you to restore any of these later if necessary.

Backup copies are squashed and stored on a user-defined backup path which may be another hard disc, a floppy disc, or network fileserver, etc.

See *Chapter 22: Making backups*.

2.15 Disc access

It is important to understand that Impact accesses data directly from the disc. While this has the advantage of allowing large files to be maintained irrespective of the amount of memory (RAM) available, it does mean that the disc files need to be kept open while record cards are displayed.

Further, since one database may access other related databases, there may be several files open at any one time, even where the record card is not displayed. Open databases are indicated by a tick on the database menu.

2.16 Closing databases

A database is closed by clicking ***Menu*** over an open card and choosing *Close database*. While databases are open, a drawer in the filing cabinet iconbar icon will be shown open. When all databases are closed, the icon will revert to its closed form.

While a database is open, a few records are held in memory. This is done to speed access to the data, and cut down on disc access and unnecessary wear and tear. This means that if there should be a power failure or other computer crash, then some modified data may be lost, as would happen with most programs.

Note that if you attempt to close a database, the program will not close it if it is currently used by another related database. It will, however, flush all data to disc to ensure that it is up to date. The *Close All* option from the iconbar menu closes all open databases. Quitting the program, from the iconbar menu *Quit* option, or from the *Task Manager*, will always close any open databases before the program terminates.

2.17 Shortcuts and hot keys

The following keyboard shortcuts are available. For these to work, Impact must know which database you are using, and so the the caret must be placed in the card.

Page up	Select previous record.
Page down	Select next record.
Home	Move to the first record (except in RISC OS 5).
Insert	Go into **entry mode** to create a new record.
Ctrl-F2	Close current card, provided no data needs saving.
F3	Save displayed data to disc.
F4	Open the **search tool**.
Ctrl-D	Insert the current date into a field.
Ctrl-F	Open the **find tool** to search in the field currently containing the caret.

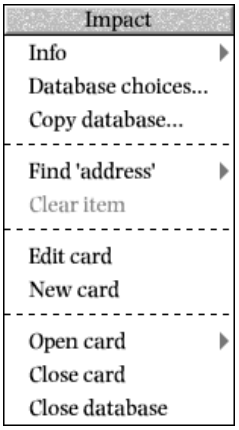
Key actions are defined using **action scripts** which are saved in the *Keys* directory within the *!Impact* application, and further keyboard shortcuts may be added by the user creating additional scripts in this directory. When any key is pressed which is not used by Impact itself, an action script in this directory is sought. If a file is found having the same name as the key's Wimp-number then this is executed. Key actions apply to all databases.

Chapter 3: Introduction to tools

This chapter describes the use of the main database tools, to select cards and to set filters and indexes etc. Many of the tools available are described in more detail in the following chapters.

3.1 The card menu

A ***Menu*** click anywhere over a record card will cause the card menu to be offered, as shown here.



The *Info* option leads to a general information box, showing details about the database.

From the *Database choices* window you can define the index, filter and field map you wish to have active when the database is initially opened, and which record is displayed. For further details see *21.3 On opening a database*.

The *Copy database* option opens a save box from which you can make a complete copy of a database, or just of its structure (excluding the actual data). This is described fully in *3.22 Copying databases*.

The *Find* option leads to the **find tool**, as described *3.18 Finding a record*. The name of the field which will be searched is shown on the menu.

Clear item is available for certain field types as a method of clearing the contents of the field, as described in *Chapter 7: Data entry*.

Selecting *Edit card* puts the card into **edit mode**, displaying the field selection window. This permits the card and toolbar to be modified as required.

The *New card* option causes a new blank card to be displayed, together with the field selection window, ready for a new card to be created for that database.

Where a database has more than one card defined, the *Open card* option leads to a sub-menu of cards available for that database, enabling you to open any hidden cards.

The last two options are used to close the card, or the whole database.

3.2 The toolbar

Each record card open will display a toolbar for use on that database. Tools are selected for the toolbar during the card editing process. It is not necessary for every card on a database to have a toolbar, but for most tools it is the only means of access.

Most tools are activated by clicking with ***Select***, when the tool icon will be displayed depressed. Selecting most tools also causes the display of a dialogue box providing full control over the tool actions, such as label printing, merging etc.

The following is a brief description of each of the above tools. Most of them are described

more fully in later chapters.

3.3 The browse tools

These allow you to browse through the database record by record. The arrows with bars select the first or last record. You can also move to the first record by shift-clicking on the **previous** arrow and to the last record by shift-clicking on the **next** arrow. All actions are reversed by clicking with ***Adjust***.

If an **index** has been selected, browsing will take place in indexed order.

If a **filter** has been selected, only those records which pass the filter tests will be displayed.

3.4 The data entry tool

Selecting this tool clears the record card ready for the creation of a new record. If a default entry has been defined, this will be displayed. For more details, see *Chapter 7: Data entry*.

3.5 The copy tool

This tool creates a new record which is a copy of the current one.

3.6 The save tool

This tool saves the current record. This may also be done by pressing the ***F3*** key.

3.7 The close tool

If clicked with ***Select***, this tool closes the current database. If ***Adjust*** is used and several cards are open on this database, only the card containing this tool is closed.

3.8 The delete record tool

Clicking this tool will delete the current record, after a warning.

3.9 The select tool

Selecting this tool allows data fields to be selected for drag-exporting. Fields may be selected in any order, clicking with ***Adjust*** to add to the initial selection. See *Chapter 18: Exporting data*.

3.10 The search tool

Selecting this displays the main search and replace dialogue box. For more details, see *Chapter 8: Search and replace*.

3.11 The merge tool

This tool gives access to the data merge dialogue box, for controlling a ‘mail merge’ operation

or report generation using *Ovation Pro, Impression* or *ImpEmail*. It also provides facilities for creating such documents. For more details, see *Chapter 10: Mail merging and formatted reports*.

Three different icons are provided for the merge tool so the user can choose the one they are most familiar with or like best, but they all link to the same merge dialogue box.

3.12 The label printing tool

This tool allows you to print database records onto sheets of labels via the accompanying *LabPrint* application. For more details, see *Chapter 11: Printing labels*.

3.13 The data import tool

This provides facilities for importing data in CSV or TSV format from a file or another application. For more details, see *Chapter 19: Importing data*.

3.14 The data export tool

The export tool allows data to be exported in CSV or a selected format, either to a data file or another application. For more details, see *Chapter 18: Exporting data*.

3.15 The backup tool

The backup tool provides a quick way of saving a compressed backup copy of the database, either to a pre-arranged file path, or by drag and drop. A saved database may also be restored using this tool, if the current version should be corrupt, or for any other reason. For more details see *Chapter 22: Making backups*.

3.16 The index, filter and field map tools

Clicking with ***Menu*** on one of these tools opens a menu of indexes, filters or field maps to choose from. If none is currently selected, clicking with ***Select*** will also open this menu; if one is selected then clicking with ***Select*** will deselect it.

Clicking with ***Adjust*** will enable you to edit the current index/filter/field map, or to create a new one if none is currently selected.

The currently selected and active index, filter or field map directly affects the function of all other tools.

For more details, see *Chapter 4: Filters*, *Chapter 5: Indexes*, and *Chapter 6: Field maps*.

3.17 The table tool

Clicking with ***Select*** on the table tool will open a table display. This will show all records consistent with any active **index** or **filter**, and will display data fields according to any active **field map**.

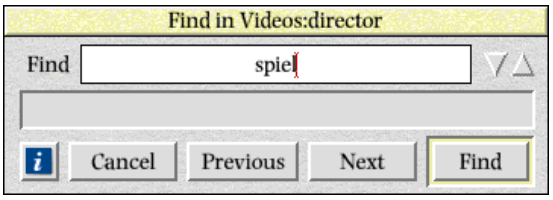
Tables are described in more detail in *Chapter 9: Tables*.

3.18 Finding a record

When a particular record is required, a search may be done. There are two ways to do this.

The **search tool** may be used to search over several data fields, and this facility also provides a global search and replace across the whole database. This is described in *Chapter 8: Search and replace*. In addition, you can search single fields using the **find tool** provided from the record card.

To use the **find tool**, ***Alt-***double-click on the required field on the card, or choose *Find* from the card menu, to display the Find window, and enter the data to be searched for.



If the field is a text type, a case insensitive search will be made for any record containing the entered text within the field. If the field is numeric or a date type, an exact match with the data will be sought, irrespective of the format used.

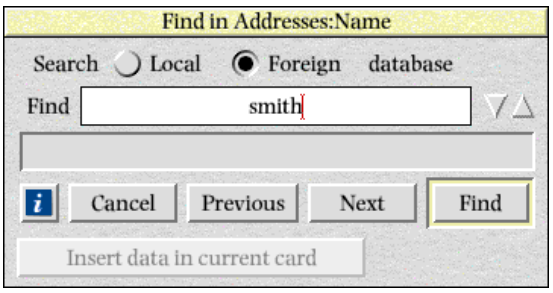
Having entered the data, the search will be initiated either by pressing the ***Return*** key, or by clicking on the *Find* button. To find further matching records, starting from the current one, click on the *Next* button, or press ***Shift-Return***. The *Previous* button will search backwards from the current one. *Find* will always search from the first record in the database or current index.

The record displayed in the card will change to show the result of the search.

The Find window remains open to facilitate further searches, and can be closed by clicking on *Cancel*. The search terms which you have used are remembered, and you can access them using the up and down arrow buttons to the right of the search term box, or the up and down cursor keys.

3.19 Finding related data

If the **Find tool** is invoked from a foreign (relational) field, the Find window will display two additional options.



The *Foreign* option is selected by default. The search will be made for a matching record in the foreign database. If a match is found, the data is displayed in the grey box below the search term, and the *Insert data in current card* button is enabled. Click on this button to enter this data in the current card. When the record is saved, the new foreign link will be formed replacing the previous one.

Selecting the *Local* option causes a search to be made on the local database as normal.

3.20 Editing a record

Whenever a data card is displaying a record, the data displayed may be edited. When this is done, the card's title bar will display an asterisk, indicating that data has been changed but not saved.

Changed data may be saved to disc by clicking on the **save tool**, by pressing ***F3*** or by pressing the ***Return*** key in a writable field (this may be disabled in the **Choices** window).

Editing and entering new data is described in *Chapter 7: Data entry*.

3.21 Closing databases

Clicking on the **close** tool, or selecting *Close database* from the card menu, will cause the database to be closed.

Note that where relational links exist between databases, Impact will open the related databases automatically. Closing a database will close all related databases, provided they are not referred to from another open database, or currently displayed on screen.

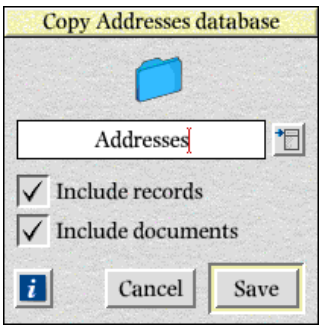
All open databases are shown ticked on the database menu, and while any databases are open the Impact icon on the iconbar will show an open drawer in the filing cabinet. To ensure that all databases are closed, and their data safe, the iconbar menu option Close all should be selected.

All databases are closed automatically when *Quit* is chosen from the iconbar menu, or the program is terminated using the Task Manager.

3.22 Copying databases

Each Impact database is stored in its own directory, usually inside the UserData directory that sits alongside the !Impact application directory. If Impact is not running, you can simply copy the database by copying its directory.

A wider range of copying options is available using the *Copy database* tool on the card menu.



This tool allows you to make a copy of a database when Impact is running. You can choose whether or not to include the records in the database copy, and whether to include any documents for mail-merging or internal file field templates.

Producing a copy of the database without including the records is useful if you have created a database and wish to share the design with another Impact user.

In most respects the *Copy database* tool is a standard RISC OS save box. You can drag the directory icon to a Filer window to make a copy of the database wherever you like. You will notice, however, a menu button to the right of the filename icon. This provides a convenient method of choosing a directory in which the copy will be saved. After adjusting the database name, if necessary, you can then just click the Save button to make the copy.

The directories offered in the menu are those which Impact examines when generating the menu of databases on the iconbar menu. By default this will be the UserData directory, but Impact

can be configured to look for databases in several different places. For more information, please see the following sections of the manual:

- *24.2 Storing databases in diverse locations* for information on how to configure Impact to include databases stored elsewhere on its menus;
- *24.4 Loading databases by unconventional means* which explains how to open a database whose directory is not included on Impact menus.

It is also worth taking a look at *24.5 Copying and renaming databases* for further advice on copying, particularly handling relational databases, backups, and mail merge documents.

The *Copy database* tool will warn you and ask you to confirm before overwriting an existing directory. It also tries hard to prevent you from overwriting a database with copy of itself, as this would destroy the data. It's possible to circumvent this protection by doing wacky things with shared drives or symbolic links, but in normal use you should be safe!

Impact also has a useful backup tool, which allows you to produce regular compressed backups of data. Users are strongly advised to take advantage of the backup tool to keep their data safe. Please see *Chapter 22: Making backups*.

Chapter 4: Filters

This chapter describes filters and how to create and edit them.

No database is of much use unless data relevant to the task in hand can be extracted easily. However, the possible range of conditions you may wish to apply to select the required data records is potentially infinite, and this often leads to the use of very complex and unfriendly query languages being employed.

Impact has been written to simplify this operation as much as possible, reducing the actions required to simple menu selections wherever possible. However, quite complex selection criteria may still be used, to enable the most demanding tests to be applied.

Filters are tests which may be applied to database records, to select a set of records required for a particular operation. Each filter specifies a single test condition on a specified field, but filters may be combined to define more complex tests.

Unlike indexes, filters do not use a list held in memory. Instead, Impact applies the filter tests as required, to decide whether to use each record as needed.

4.1 Creating and editing filters

To create a new filter from scratch, you must first deselect any currently active filter by clicking with ***Select*** on the **filter tool** if this is depressed.

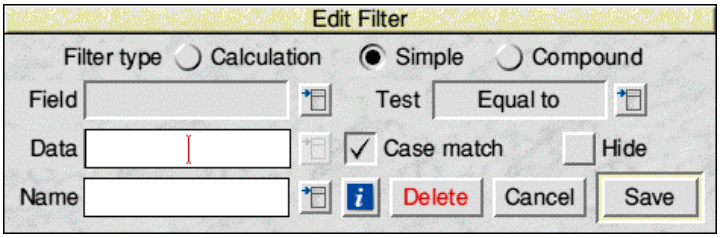
If you wish to edit an existing filter or create a new one by copying an existing filter definition, that filter must first be selected by clicking with ***Menu*** on the filter tool. If the current record does not pass that filter test, Impact will issue a warning to this effect. You can ignore this if you wish to edit the filter.

In all cases you should then click with ***Adjust*** on the filter tool to open the Edit Filter panel.

There are three types of filter: simple, calculation and compound. A **simple filter** applies a single test to a single data field. A **calculation filter** uses an expression to determine the result of the filter test. A **compound filter** combines two existing filters, defining a logical operation between them, such as AND, OR etc.

4.2 Simple filters

Simple filters define a single test applied to a specified field. To define a simple filter, select the *Simple* option button.



The field to be tested can be selected by clicking on the *Field* menu icon and selecting from the menu offered.

Next the filter test should be selected by clicking on the *Test* menu icon, and choosing from the range of tests offered. Each test refers to the data to be entered in the *Data* box. For example, if *Greater than* is selected, then the test will pass if a given record has a field value greater than the value entered in the *Data* box.

If the *Range* test is selected, then the *Data* box will be split into two, and a minimum and maximum value for the test may be entered. The test will pass if the field value is equal to or greater than the first record, and less than the second, i.e. a range of 100-200 will select all

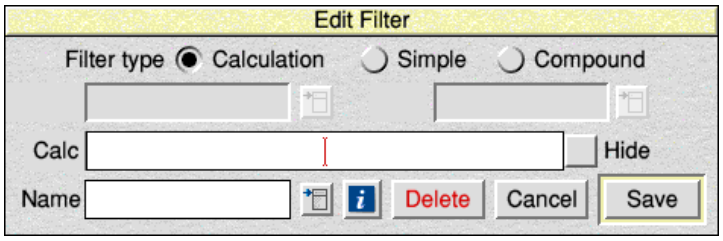
values from and including 100, up to and including 199.

If the *Partial match* test is selected (text fields only), then the test will pass if the field contains the sequence of characters entered as data.

Text field tests may also be carried out in case sensitive mode, by selecting the *Case match* option. If upper and lower case characters are to be treated as equal, this option should be off.

4.3 Calculation filters

Calculation filters use an expression to determine the result of the filter test. To define a calculation filter, select the *Calculation* option button.



The expression used in a calculation filter may include field names from any database. When executed, the result is converted to a **number** data type. Any non-zero value will be taken as true, and result in the current record passing the filter test. A value of zero (false) will cause the record to fail the test. See *Chapter 16: Calculations* for details of how to construct calculations.

A test could be made to compare the contents of two fields, and pass the record if they are equal, without needing to know the actual field contents. Or the test could include data from another database, to filter one database's records according to the selected record in another. Another use may be to select records according to a date field, applying a test relative to the current date. Say all records dated before today, or more than 8 weeks old.

Testing flag fields

One particular case where a calculation filter is useful is for testing whether one of a set of flags is ticked. You can have several tick-boxes stored in a single flag field and the value stored in the database is built up like a binary number (see *16.15 Using flag fields* for details).

Supposing you have a set of flags with the field name 'Status' and you want to test whether the second flag in this set is ticked. The calculation expression you would need to enter in the filter definition is:

```
Flag(Status,2)
```

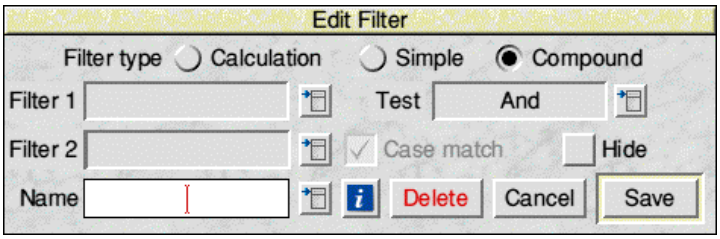
If you wanted to filter for records where the second flag was **not** ticked, you would use this expression:

```
!Flag(Status,2)
```

These expressions both use the Flag function which is described fully in *16.13 Miscellaneous functions*.

4.4 Compound filters

It is possible to combine the effects of two existing filters, so that both tests will be applied, and their results combined in some way to decide the final result. The two filters used may be any combination of simple, calculation or other compound filters. To define a compound filter, select the *Compound* option button.



It is now possible to select the two filters to be combined to form the compound filter, and the *Test* menu used to select from the range of tests specifying the way in which the two filters are to be combined.

These are AND, which will require both filters to pass, OR requiring either or both to pass, NOR which will assume a pass if both filter tests fail, and XOR (exclusive or) which will assume a pass if one and only one of the two filter tests pass.

4.5 Creating the filter

All that is now required to create the filter is to enter a suitable name and click on the Save button. Once a filter is created, its name will appear on the Filter menu from the filter tool.

If the displayed filter name is the same as an existing one, you will be asked to confirm that you wish to redefine the filter. If that name has not been used before, a new filter will be created.

4.6 Deleting a filter

Having selected a filter as described above, it may be completely deleted by clicking on the Delete action button.

Note however, that you cannot delete a filter which is used in an index definition, or one which is used as part of a compound filter. An error message will be given if this is attempted.

If you have made changes to a filter, clicking the Cancel button will close the filter panel ignoring all changes made.

4.7 The Hide option

If this option is selected, the filter will not appear on the filter tool's menu. This may be found useful where a simple or calculation filter has been defined whose only purpose is for inclusion in a compound filter, and will not be used on its own.

To edit a hidden filter, open the edit panel first, then select the filter from the *Name* menu, which will now show all filters including the hidden ones.

4.8 Filter tests

This section describes the filter tests in detail.

Simple filter tests

These are the actual tests carried out on each record. The table gives the condition for the test to pass, that is, for the record to be used.

Test	Operation
Equal to	Field content equal to test data
Not equal to	Field content not equal to test data
Greater than	Field content greater than test data
Less than	Field content less than test data
Range	Field content equal to or greater than lower limit, but strictly less than upper limit
Partial match	Field content includes test data (Text data types only)

For text fields, matching may be case sensitive or insensitive according to option setting. Matching is otherwise done according to the ASCII code sequence.

The *Partial match* test will pass records where the text given is found within the field data. For instance, if the test data is Smith, and the field contains Mr John Smith, then the record will pass the filter test.

Partial match is only available on text fields.

Compound filter tests

These are used to define the relationship between selected filters when creating a compound filter. The condition required for the compound filter to pass a particular record is given in terms of the pass state of the two sub-filters.

Test	Condition for filter pass
AND	Both filters must pass
OR	Either or both filters must pass
XOR	One and only one filter must pass
NOR	Neither of the filters must pass

4.9 Default settings for filter selection

You can choose whether a filter will be activated when the database is first opened. Impact can remember the filter which was active when you were last using the database, or a specific filter which you wish to have active when opening the database. For example, if you had a database of current and former members of a club, you might wish the database to open showing the current members only.

These options are explained fully in *21.3 On opening a database*.

Chapter 5: Indexes

This chapter describes indexes, and how to create or modify them.

An index is a list of field entries held in memory in sorted order, allowing fast, sorted access to the data records on disc. As records are entered or deleted, all index lists are kept up to date automatically. As this does not affect the order of records on the disc, it is quick, and also allows many indexes to exist at the same time, and to be chosen when needed with a simple menu selection.

Many named indexes may be created for the same database, and each may be sorted on any number of fields. Once created, an index will automatically appear on the menu available from the index tool.

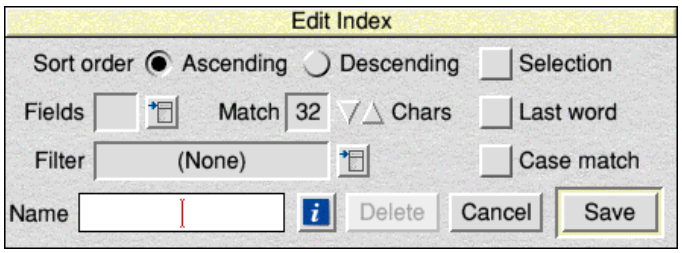
Impact loads all indexes for a database into memory when the database is opened, in order to effect fast access, and this may consume a large amount of memory (RAM). This should be borne in mind while creating indexes, if your computer does not have much spare memory.

5.1 Creating and modifying indexes

To create a new index from scratch, you must first deselect any currently active index by clicking with ***Select*** on the **index tool** if this is depressed.

If you wish to edit an existing index or create a new one by copying an existing index definition, that index must first be selected by clicking with ***Menu*** on the index tool.

In all cases you should then click with ***Adjust*** on the index tool to open the edit index panel.



When first displayed, the panel will display details of the currently selected index, if any. This index may be edited and re-saved, either using the same name to modify the existing index, or with a new name to create a new index.

5.2 Selecting the sort fields

The first thing to do is to select the field or fields on which the sort should be made, and this is done by selecting from the *Fields* menu. If you do this with the ***Adjust*** button the menu will stay open allowing several selections to be made. The fields will be numbered on the menu to show their order of priority. A selected field may be de-selected by clicking on it a second time.

A second or later field will only be used in the sort where the first field has two or more matching records. While it is quite possible to select many fields, the field data for all selected fields will be included in the index, which could become unduly large. Care should be taken to include only those fields which are really needed, according to the application. In most cases it is unlikely that more than two fields would be needed.

5.3 Text fields

If the chosen field is a **text** type, then the space taken by the field data within the index could be large. If it is unlikely that any two records will match over more than the first few characters, it is useful to restrict the number of characters used during the sort, and therefore minimise the memory space needed.

This may be done by setting the number displayed in the *Match* box, using the adjuster buttons provided. The maximum number of characters is 32. The match setting has no effect for numeric fields, such as numbers or dates, etc.

5.4 The last word

If this option is selected, the sort will be made using the last word found in a text field, and this may be conveniently used to sort names on the surname only. The *Match* setting is still used.

5.5 Case matching

It is also possible to choose whether text comparisons are to be made having regard for upper and lower case characters. If the *Case match* option is selected, then upper case (capitals) will be regarded as filing before lower case characters, and will be sorted accordingly. With this option off, upper and lower case characters will be considered to be equal.

5.6 The sort order

Records may be sorted either in ascending or descending order, by selecting the appropriate option. This setting will apply to all fields chosen.

5.7 Using a filter

Normally, an index will contain all records in the database. However, it is possible to form an ordered sub-set of data records, by selecting a suitable **filter**. This may be done by clicking on the *Filter* menu icon, and selecting from the displayed menu. Filters are described more fully in *Chapter 4: Filters*. If a filter is selected in the index definition, then only records which pass that filter will be indexed, which saves memory.

It is possible to define filters such that whether records pass the filter depends not just on the record data but on some external characteristic, such as the time or the computer which is being used. For example, you might have a filter which shows people who have birthdays this month, or a filter which detects file objects stored on the local hard disc. Indexes are updated when a record is edited, but will not be updated just because the time has changed or because you are using a different computer. You must therefore be careful not to choose such filters in index definitions.

5.8 Manual record selection

It is possible to create an index containing records which have been manually selected, rather than have the index contents selected automatically. This is done by selecting a set of individual records from a **table display** (use the **table tool** as described in *Chapter 9: Tables* to open a table, and select the records you require).

Having made the required record selections in the table display, the *Selection* option should be chosen in the Edit Index panel. The other settings (except for a filter selection) may then be made as described above, to choose the field(s) to be sorted on, etc.

We do not recommend using manual selection at present for anything other than short-term storage of selections, as it is very difficult to edit an index based on a selection. Probably a better approach is to add a **flag** field to the database card which shows whether that record is required in the index, and then use a filter on this field in the index definition.

5.9 Creating the index

All that is further required to create the index is to enter a suitable name in the *Name* box, and

then click on the *Save* button. If the name is that of an existing index, you will be asked to confirm that you wish to replace this with the new definition. Index creation may take a few moments if the database is large.

Once created, the new index will appear on the index menu available from the index tool, and will be automatically updated when records are added or deleted.

5.10 Deleting indexes

Existing indexes may be deleted when no longer required, and this will save some disc and memory space. To delete a named index, click the *Delete* button. Confirmation will be requested before the index is deleted.

5.11 Default settings for index selection

You can choose whether an index will be activated when the database is first opened. Impact can remember the index which was active when you were last using the database, or a specific index which you may wish to apply when opening the database.

These options are explained fully in *21.3 On opening a database*.

Chapter 6: Field maps

This chapter describes **field maps**, or selections, and how to create them.

Field maps consist of ordered lists of data fields, which are given a user defined name. They are used whenever a selection of fields is needed, for example when printing labels.

In addition to selecting the fields in the required order, you can also choose which fields are to be concatenated onto the end of the previous field, the two then being treated as a single field. This feature is useful. for example, to cause forename and surname fields to be printed on the same line of a label.

It is also possible to specify a character or characters to be inserted between the two concatenated fields (often a single space).

6.1 Creating and modifying field maps

To create a new field map from scratch, you must first deselect any currently active field map by clicking with ***Select*** on the **field map tool** if this is depressed.

If you wish to edit an existing field map or create a new one by copying an existing field map definition, that field map must first be selected by clicking with ***Menu*** on the field map tool.

In all cases you should then click with ***Adjust*** on the field map tool to open the edit field map panel.

If you are editing an existing field map, its definition will appear in the panel.

6.2 Field selection

There are two ways to select the required fields: by selecting them from the field selection menu or by selecting them from the card.

Using the first method, click the *Field selection* menu icon and select the required fields from the menu. If you do this with the ***Adjust*** button the menu will stay open allowing several selections to be made. The fields selected will be numbered on the menu to show the selection order. A selected field may be de-selected by clicking on it a second time.

To select fields from the card, click on the *Show field selection* option button. When you do this, any already selected fields will be shown selected on the record card. Further selections/ de-selections may be made on the card using the Select and/or Adjust mouse buttons as usual. The field order will be the order in which you select them on the card, and this will be reflected on the field menu when displayed.

6.3 Concatenating fields

This feature allows you to define which fields are to be concatenated when printed or exported. A field flagged as concatenated will be joined onto the end of the previous field, after optionally inserting specified characters between them. Any characters to be inserted should be entered into the writable icon labelled *Concatenate with*, for example one or more spaces.

Fields for concatenation may be chosen from the concatenate fields menu. This menu will show the fields selected, in order, and provides a useful check that the fields have been selected in the

desired order. Note that the first field will be shown faded, as in this case there is no previous field to concatenate it with.

6.4 Creating the map

To create the field map, enter a suitable name in the *Name* box, or edit the existing one if required, and click the *Save* button.

If a field map with the same name exists, you will be asked to confirm if you wish to redefine it. If the name is unique, a new map will be created.

6.5 Deleting a field map

Having chosen an existing field map, it may be deleted by clicking on the *Delete* action button. A warning will be given before the selection is deleted.

6.6 Default settings for field map selection

You can choose whether a field map will be activated when the database is first opened. Impact can remember the field map which was active when you were last using the database, or a specific field map which you may wish to apply when opening the database. For example, if you have a database with a large number of fields per record, you might have a convenient selection of fields defined in a field map suitable for displaying the records in a table. Having this field map active each time the database is opened would allow you to open a table and jump quickly to the record you require.

These options are explained fully in *21.3 On opening a database*.

Chapter 7: Data entry

This chapter describes how to create new records using the data **entry tool**, and how to modify existing records.

7.1 Modifying existing data

As normally displayed, a data card will be displaying a particular record. Any field may be readily amended, and the record re-saved.

When a field has been modified, the card title bar will show an asterisk * after its title, in conventional manner. This is a reminder that the changes made have not yet been saved to disc.

In general, modified or new data must be saved to disc by clicking on the **save tool** on the toolbar, but this depends on the options chosen in the **Choices** window described elsewhere.

Modifying existing data is similar to entering new data, which is described below for each data field type.

7.2 Using drag and drop

Data may be dragged from one Impact field to another, within one card or between different cards or different databases. A simple ***drag*** will copy the data, replacing the contents of the field in which it is dropped. ***Shift-drag*** will move the data, clearing the field from where the data was dragged. ***Ctrl-drag*** will append to the existing field data instead of replacing it. If you have partial text selection enabled (see *21.11 Text selection*) then these actions are modified when text fields are involved as you will not necessarily be copying or moving the whole contents of a field. See *Appendix 2: Mouse and key actions in record cards* for a table showing the difference made by enabling text selection.

Text or images may also be dragged into an Impact record card from another application. For example, a name or address could be selected in an e-mail or wordprocessing document, and dragged directly into a field on the data card. This will replace the existing field data, or, if it is a text field and you have enabled partial text selection, insert it at the pointer. If the other application does not support direct drag and drop, you may need to select the text and then drag from a *Save* box rather than directly from the document.

The interpretation of the imported text will depend on the field type the text is imported into. For example, if importing into a date field, Impact will do its best to interpret the text as a date.

7.3 Entering new data

To make a new record in a database, click on the data **entry tool**. This action will clear the card and place the caret into the first writable icon. You can also use the ***Insert*** key short-cut.

When this is done, any default data entry defined for the field will be inserted after the card has been cleared. This data may be kept or edited as required. (Setting default data is described later in *Chapter 12: Creating a database*.)

In this mode, successive records may be created; whenever data is saved to disc by pressing ***Return***, the card will be automatically cleared ready for the next record. To exit this mode, de-select the entry tool by clicking on it again.

Data entry methods vary between the different data field types, so each will be described separately.

7.4 Writable fields

These include **text**, **number** and **date** field types, which all appear as a white writable icon. Data is entered by first placing the caret into the field, if not already there, by clicking the Select mouse button over the field. Alternatively, the up and down ***cursor keys***, or the ***Tab*** key, may be used to move the caret between fields, in the standard desktop manner.

With the caret in the writable field, data may be typed directly into the field, and normal editing may be carried out using the ***Delete*** key, etc.

7.5 Date fields

Dates may be entered in almost any format, regardless of the field's display format. (How to define the date format for a field is described later in *14.13 Date/time fields*.)

When the record is saved, the date will be validated, and any unrecognised or illegal dates will be rejected.

Any date between 1900 and 2247 is acceptable. If you enter a year in abbreviated two digit form, numbers from 20-99 will be assumed to be in the 20th century and those from 00-19 will be assumed to be in the 21st century, i.e. 19 will be assumed to mean 2019 and 20 will be assumed to mean 1920. To enter any dates before 1920 or after 2019 you must use all four digits. To avoid confusion it probably makes most sense to type in all four digits for any year you require.

Note: if you wish to store dates outside the range 1900-2247 you will have to use a text field.

7.6 File fields

This field type appears as a click button and displays its legend in black when there is a link to a file, in grey when there is no link, and in red when the linked file cannot be found.

Clicking on such a field is equivalent to double clicking on the file in its Filer window, and a shift-click on the button would correspond to a shift-double-click on the file itself.

There are two types of file field:

- internal file (formerly called "Notes"). The file is stored inside the database structure.
- external linked file. In this case the field stores the full pathname of the file, which may be anywhere on your computer.

External file fields

You can drag any file, directory or application from a Filer window and drop it on this field to create a link. When this is done, the displayed legend will turn green, indicating that the data has been changed but not yet saved, and the card's title bar will show that the card data has been modified (by the addition of an asterisk).

Note that Impact simply stores a note of the location of the file, not a copy of the file itself, so the file may be altered later without any need to amend the Impact record. If the file has been moved or deleted since the record was edited, the button's legend will be red to indicate that the file could not be found. If you click on such a button, you will get an error message which tells you what file it was linked to.

To avoid irritating repetition of errors when a disc cannot be found, Impact keeps a note of which discs cannot be found and does not attempt to access the same disc again. If the disc becomes available later but the file button is still coloured red, you need to click on the button

to get Impact to check for that disc again.

Although Impact appears to store the full pathname to the file, in fact the disc name is stripped off the front if the file is kept on the same disc as the database. This enables Impact to behave correctly if you open the database across a network from another computer. See *24.3 Using Impact over a network* for more details.

To remove a file link see *7.11 Clearing fields*.

Internal file fields

These are held as individual files within the database, and are created automatically by Impact.

You can store any content in an internal file field simply by dragging the file and dropping it onto the button on the card. Files do not have to be confined to text or document files: you can store images, audio or even computer scripts.

When you drag a file and drop it on an internal file button, you may be asked whether you want to overwrite the existing data in the field. This is a precaution against users dropping content in the wrong place, as it is not obvious that the content of the field has changed as it would be for other fields. The new file is stored away inside the database, but is only fully attached to the record when you save the changes to record as a whole, so you also have the opportunity of navigating to another record and discarding changes.

If you open an existing file to view it in an editor, you can re-save from within the editor application using its existing file path, and the content is re-saved within the database. As the user can see what changes have been made via the editor, the changes are immediate and do not depend on saving the record card. You can, of course, save a copy elsewhere, or export the file into another application.

If the button legend is shown in grey, then clicking the button will create a new empty file, after prompting you to confirm that you wish to do so. If the file field has a template provided, it will be copied and used as the basis of the new field. The new file will then be loaded into an appropriate editor.

A template may be of any file type, and may be empty or have some default content as you wish. If there is no template file, Impact will create an empty plain text file when needed. However, you can supply another type of template file, say an *Impression* document, if preferred. This is fully described later: see *14.14 File and image fields*

Different records may have files of different types stored in an internal file field if you wish: they do not all have to follow the same template. You will be warned, when dropping a new file on a field, if the filetype of the new file does not match the template for that field, and you can choose whether to proceed or not.

To remove internal files see *7.11 Clearing fields*.

7.7 Image fields

An image field is simply an internal or external file field where the image is displayed in the card instead of showing a button. The image is scaled to fit the space available, and may be opened in another application by double-clicking or dragged and dropped into other applications.

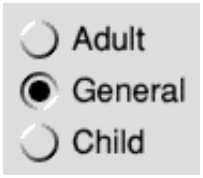
If Impact is unable to display the image itself, a filetype icon will be displayed for the appropriate file type instead. Impact can display sprite, JPEG and Draw files on all current versions of RISC OS, and on RISC OS Select a much wider range of file types is supported.

If the image is an external file which cannot be found, a red warning triangle will be displayed instead. A question mark in a red circle shows that the disc is inaccessible.

To remove an image, see *7.11 Clearing fields*.

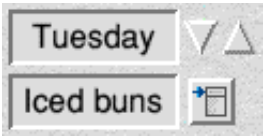
7.8 Flag and option fields

These types are set by simply clicking the appropriate button. As with many other applications, radio buttons can be deselected by clicking with ***Adjust*** if you wish none of them to be set.



7.9 Menu and browser fields

These fields are set in the normal desktop manner, either by clicking the browser arrow icons, or clicking the menu icon and making a selection, as appropriate. Note, if you wish to clear the value you must have a blank option available on the menu.



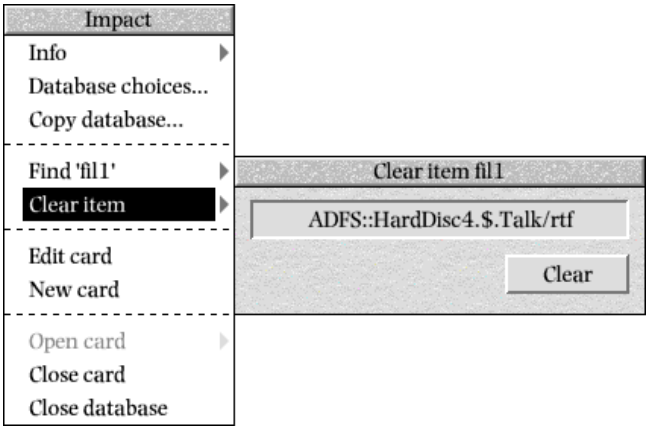
7.10 Saving the data

To save any changes it is necessary to click on the **save tool** or press ***F3***. By default, pressing the ***Return*** key in any field will also save the changes. If you prefer, in the **Choices** window you can make the ***Return*** key move to the next field instead of saving the record.

If you attempt to deselect the entry tool or browse to a new record while there are unsaved changes displayed, the program action is dependent on the current settings in the Choices window available from the iconbar menu, as described in *Chapter 21: Choices*.

7.11 Clearing fields

File and image fields may have their contents cleared by choosing *Clear item* from the **card menu**. Follow the arrow to the right to see a brief description of the field data (in the example below, the file name) and click on *Clear* if you wish to remove the object. For internal files, the file will be deleted; for an external linked file it is just the link that is removed.



When this is done the card will be flagged as edited, and will need saving to make the change permanent. File fields will have their button text changed to red to indicate that the change is not saved. When the card is saved, the button text will change to grey to indicate that this object contains no data.

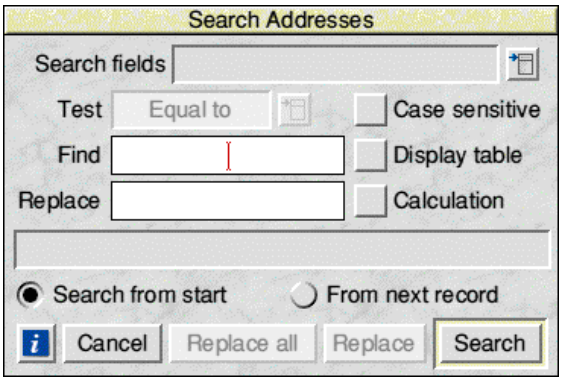
Chapter 8: Search and replace

This chapter describes the use of the **search tool** to carry out global search and replace operations. It is also possible to make changes to many records at the same time using the table editing features. See *9.13 Editing several selected records together* for more information.

This tool will search through the database for records matching the data you supply, and will optionally replace the found fields with a new value or with the result of a supplied **calculation**. Calculations are used in other parts of Impact, and are fully described in *Chapter 16: Calculations*.

Using this tool it is possible, for example, to switch off flag fields which are on, multiply a number field by some factor (or perform a more complex calculation), or add a given number of days to a date, etc. This may be done with an index or filter active, to control which records are affected.

8.1 The search window



This window is displayed when the **search tool** is selected or ***F4*** is pressed.

The fields to be searched may be chosen from the field selector by clicking on the *Search fields* menu icon. The search can be done over several fields, but it is only sensible to search over similar field types. For this reason, some fields will become faded following your initial selection.

The test to be used on the fields must then be selected from the *Test* menu. These tests are the same as those described for **simple filters** in *4.8 Filter tests*. For text fields, you can also choose to apply the test in a case sensitive manner if required.

The data to be searched for should then be entered in the *Find* box. Where text is being searched for, the *Case sensitive* option may be selected if required.

If searching on a **flag**, **option**, **browser** or **menu** field, the data should be entered in text form. That is, you should enter the text shown on the card for that item. For these types of field the matching will always be case insensitive.

There may be more than one flag contained in a single flag field, and you can search for records with a combination of flags ticked by separating the flag texts by commas in your search string. A database of books might have flags labelled ‘Paperback’, ‘Illustrated’ and ‘Non-fiction’. If you wished to retrieve records for illustrated non-fiction you would enter the search string

```
Non-fiction,Illustrated
```

Note the order of the flag texts does not matter. However, when searching for flags the search tool will only find those records which exactly match the flags listed, so in the example above those records which have ‘Paperback’ selected will **not** be retrieved, even if they are also illustrated non-fiction.

8.2 Entering replacement data

Simple replacements

The simplest way to replace the contents of a field is to type the text, number or date you wish to use in the *Replace* box. Ensure the *Calculation* option is not selected.

For option, flag, menu and browser fields, you should type the text of the item which you wish to have selected. To select more than one tick box in a flag field, separate the items by commas in the same way as the example above.

Using calculations in replacements

For more complex replacements, data may be entered in the form of a **calculation** by ticking the *Calculation* option. Calculations are fully described in *Chapter 16: Calculations*, but some examples are given here.

Suppose you have a *Price* field in which you wish to increase all your prices by 20%. You could search for *Price* not equal to 0, and enter the replacement calculation

```
Price*1.2
```

If you wish to extend all your current members' subscriptions by four months, you could search for *ExpiryDate* greater than today's date and enter the replacement calculation

```
AddMonths(ExpiryDate,4)
```

Suppose that in your membership database you have options to indicate the type of membership held and you only wish to extend the subscriptions for those whose *Type* is marked *Personal*, you could use the replacement calculation

```
AddMonths(ExpiryDate, Type=="Personal" ? 4 : 0)
```

This would add four months when the expression `Type=="Personal"` is true, and no months otherwise.

If the membership type is a **flag** field rather than an option, you must use the numerical value of the flag settings instead. Suppose *Type* has three flags marked *Complimentary*, *NoMailing* and *Personal*. If you wish to extend the subscription for all personal members (the third flag in the field) regardless of the other flag settings it is easiest to use the **Flag** function:

```
AddMonths(ExpiryDate, Flag(Type,3) ? 4 : 0)
```

If you wish to make a replacement in a **flag**, **option**, **menu** or **browser** field using a calculation, then the calculation may either yield a text string which will be matched against the field's item texts, or a number which will indicate which item(s) should be selected. For menus and browsers the items are numbered from zero; for options zero indicates that nothing is selected and the first item is numbered 1.

For flag fields, the number is bit-significant to indicate which flags are selected, i.e. 1 indicates the first alone, 2 the second alone, 4 the third alone, 3 the first and second, 5 the first and third, etc. To set the third flag of a field called *FlagField* and leave the others as they are, you could use the calculation

```
FlagField | 4
```

More information on flag field values may be found in *16.15 Using flag fields*, and the **Flag** function is described in *16.13 Miscellaneous functions*.

8.3 Starting the search

When first displayed, the *Search from start* option is selected, so that clicking the *Search* button, or pressing ***Return*** in either of the writable boxes, will begin the search from the start of

the database, or from the start of the index if one is active.

If a matching record is found, it will be displayed in the card, and the *From next record* option will be automatically selected.

8.4 Replacing the data

If a calculation has been entered, clicking the *Replace* button will cause the field to be replaced. Note that if more than one field is searched, then only the first field which matches the data will be replaced.

Since the *From next record* option is now selected, further clicks on the *Search* button will cause the search to be resumed from the next record. Again, the *Replace* button may be used if required.

During this action messages will be displayed in the window's display box to keep you informed of progress.

Where the *Partial match* option has been used on a text field, the replacement text will be used to replace the matching sub-text, not the whole field content.

8.5 Automatic replacement

When you are happy that the replacement operation is being performed correctly, clicking on the *Replace all* button will cause Impact to search and replace continuously through the database.

The number of records altered will be shown in the display box.

8.6 Tabulating finds

If the *Display table* option is selected, the database will be searched throughout (taking note of any filters which may be active) and the matching records will be displayed using the **table display** facility. You cannot use *Replace* if you are displaying a table of the search results.

If this selection of records is to be saved for later, you can use the *Selection* option available in the **Edit index panel**. (See *Chapter 5: Indexes*).

Chapter 9: Tables

Tables allow you to view and manipulate several records at once. They offer an easy way to locate a particular record, to select several records for output, to print or delete multiple records and to find duplicate records in a database. You can also edit much of the data displayed in a table direct, without having to call up the record card.

9.1 The table tool

Clicking with ***Select*** on the table tool will open a table display. This will show all records consistent with any active **index** or **filter**, and will display data fields according to any active **field map**.

Each record in the table is displayed on a single line with the field values in columns. You may change which record is currently displayed in the data card simply by clicking with ***Ctrl-Select*** on that record in the table.

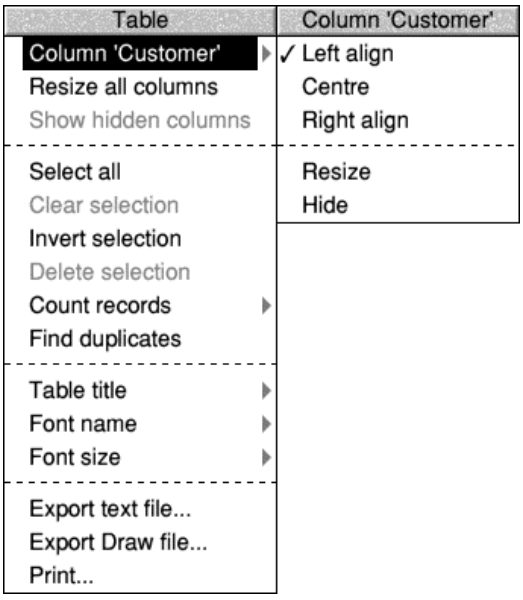
Several tables may be open on a database at once. Individual tables may be closed by clicking on the window's close icon, or all tables may be closed by clicking with ***Adjust*** on the table tool.

It is possible to make multiple record selections from the table display, and form an index containing the selection. This is fully described in *Chapter 5: Indexes*. Selections from a table may also be used for exporting records, printing labels, and report creation as described in the relevant chapters.

Note that the lines of any multi-line text or note field displayed in a table will be shown as a single line concatenated with commas. Blank lines at the start and end of such fields will be omitted from the display.

9.2 The table menu

Clicking with ***Menu*** over a table display will open the table menu.



This offers several actions associated with record selections, as well as the opportunity to reformat and print the table, or to export it in various formats.

9.3 Aligning content of columns

When you open a table, the contents of each column will be aligned left, right or centred

according to the setting for that field in the record card. You can change the alignment from the *Column* submenu of the table menu.

9.4 Resizing table columns

Column widths can be adjusted by placing the mouse over the line which separates adjacent column headings. The mouse pointer will change shape and you can drag the column boundary to the width of your choice. Alternatively, double-click on the boundary to resize the column to fit the text in the table exactly. You can also do this by selecting *Resize* from the *Column* submenu of the table menu. You can resize all columns at once by choosing *Resize all columns* from the main table menu.

Sometimes you may have a column of data where the heading is much wider than the data in the column. By double-clicking with ***Adjust*** on the column boundary the column will be resized so that all the data will fit in the width but the heading will be truncated.

There is an option in Impact's **choices** (see Chapter 21) which will automatically open tables with columns the right width to fit the data. This is turned off by default, as with a large database on slow hardware this may cause the table to appear more slowly than desired.

9.5 Hiding and reordering table columns

If you do not wish to view or print all the fields shown in the table, you may hide columns by choosing *Hide* from the *Column* submenu of the table menu. *Show hidden columns* will reveal all hidden columns again.

The order of columns may be changed in the table, by dragging the column heading sideways until it moves to the desired location. If you press *Escape* in the middle of a drag the column will be returned to its original position.

These operations only affect the table which is open at the time. To save the columns displayed and their order for future use you should define a **field map** (see Chapter 6).

9.6 Changing the title of the table

The heading in the table window's titlebar is set by default to show what filter, index and field map were in force when the table was opened (if opened by the table tool) or what was searched for (if opened as the result of a search). This can be altered from the table menu. The same title will be used if you print out the table.

9.7 Changing the table font and size

The font and size of type to be used in the table are selectable from the table menu. The desktop font will be used by default for the first table you open after running Impact. After selecting a different font name or size, tables opened subsequently will use the new setting.

9.8 Selecting records in a table

Selecting a field in the table display by clicking on it with ***Select*** will cause that record to be displayed in red, and any previously selected records in any table which is open will be deselected. Only one table window may have records selected at once. Clicking with ***Adjust*** will add to an existing selection, as with a filer window. You can also drag with the mouse to select several records and adjust-drag to alter the selection.

The *Select all* option causes all records in the table to be selected, while *Clear selection* will remove any selections made. *Invert selection* causes any record which was selected to become unselected, and all those which were not selected before to be selected instead.

The *Delete selection* option will delete all selected records from the database. This option should be used with care. A warning will be given before this action is carried out, to give you the chance to change your mind.

The *Count records* option will display a box showing the total record count, and the number of selections made.

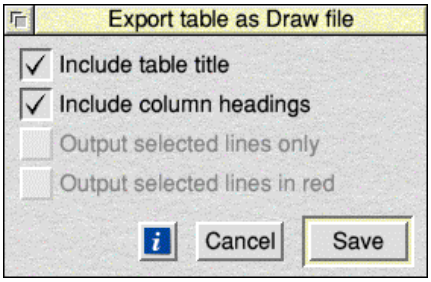
The *Find duplicates* facility can only be used if an **index** was active when the table was opened. It causes all but the first of any duplicate entries in the index to become selected, and is useful for finding duplicated addresses, etc. Any records which are still wanted may be unselected by clicking them with ***Adjust***, and the unwanted duplicates can then be deleted using the *Delete selection* option.

9.9 Exporting the table as a text file

Having opened a table and rearranged or hdden columns, you can export the data in one of two ways. The *Export text file* option opens the **Export tool** directly, rather than via the toolbar. For more information on the facilities of this tool, see *18.2 Using the export tool*.

9.10 Exporting as a Draw file

The second export method provided allows you to produce a Draw file of the table content which can be used in any application which can handle Draw files. The dialogue box allows you to choose whether the column headings and a brief description of the table content will be included in the Draw file. You may also select whether the whole table or just the selected rows will be exported.

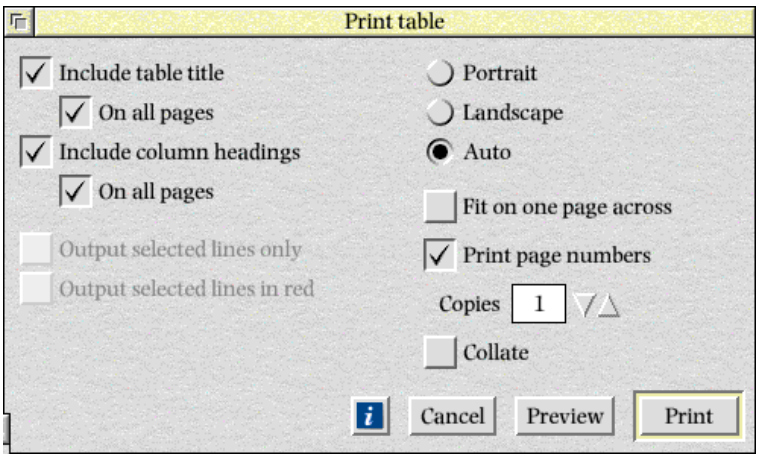


The Draw file will follow very closely the style of the table display currently on view in the table window, particularly in the style and size of the fonts and the widths of the columns. If you do not want any data truncated because it is too wide for the column, remember to use the menu option to *Resize all columns* as described in *9.4 Resizing table columns*.

9.11 Printing a table

Choosing *Print* will open the print dialogue box, to allow you to choose settings before printing to your current printer. A large table may be split over several pages, and you have the option of choosing whether each new sheet should repeat the table title (as shown at the top of the window) and the column headings at the top.

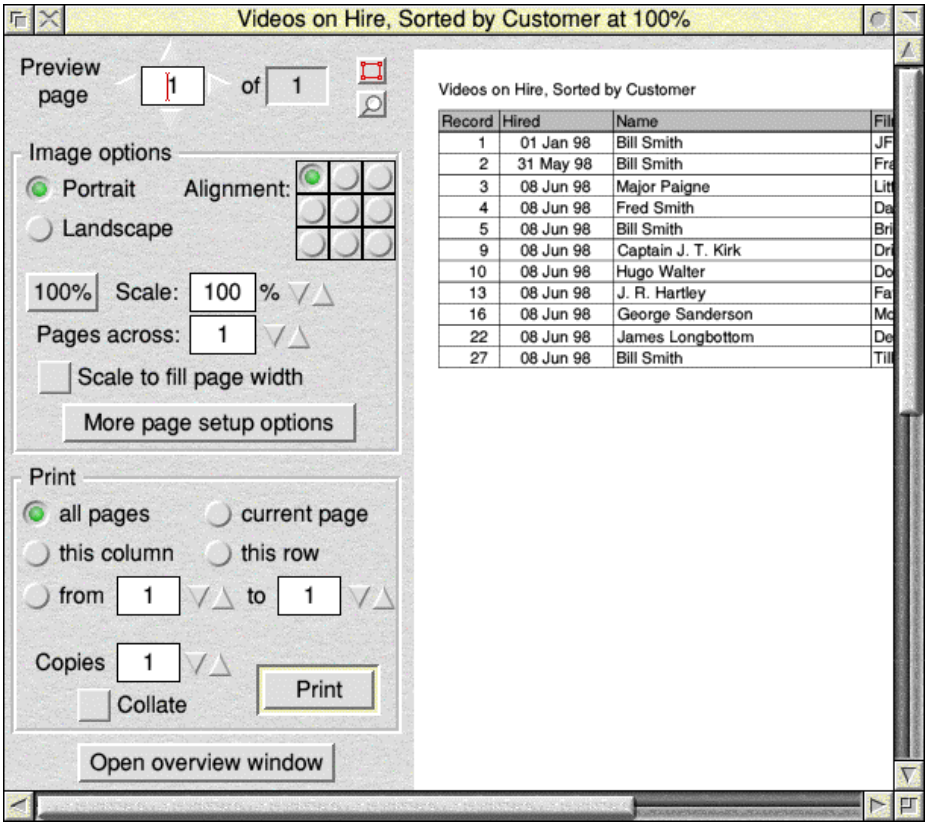
If your table has rows selected, you may choose whether to print all rows, or just the selected ones, and whether to print the selected lines in red in the same way that they appear in the window.



By default the orientation of printing will be chosen automatically to minimise the number of sheets of paper required at a scale which is not much less than 100%. If you want to force the printing to be landscape or portrait you can use the radio buttons provided.

If the table is very wide, Impact may use more than one sheet across to print all the material without having to scale down the text size unduly. The breaks between pages will coincide with the column boundaries where possible. You can force the printout to be scaled to fit a single page width if you like.

While the most useful print settings can be controlled using this dialogue box, for finer control you can use the *Preview* button. This will show the table using *DrawPrint* where you can set margins, control the scaling and page breaks, and can even crop the output if desired.


Videos on Hire, Sorted by Customer at 100%
Preview page 1 of 1
Image options
Portrait
Landscape
Alignment:
100% Scale: 100 %
Pages across: 1
Scale to fill page width
More page setup options
Print
all pages
current page
this column
this row
from 1 to 1
Copies 1
Collate
Print
Open overview window
Videos on Hire, Sorted by Customer
Record Hired Name
1 01 Jan 98 Bill Smith
2 31 May 98 Bill Smith
3 08 Jun 98 Major Paigne
4 08 Jun 98 Fred Smith
5 08 Jun 98 Bill Smith
9 08 Jun 98 Captain J. T. Kirk
10 08 Jun 98 Hugo Walter
13 08 Jun 98 J. R. Hartley
16 08 Jun 98 George Sanderson
22 08 Jun 98 James Longbottom
27 08 Jun 98 Bill Smith

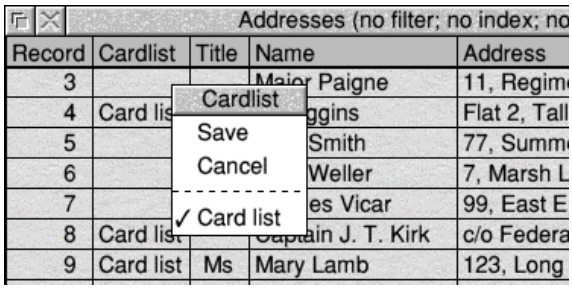
For a full explanation of the options, see *DrawPrint*'s own manual, which you can open using the *Help* option from the *DrawPrint* menu or by pressing ***F1***.

9.12 Editing data within the table

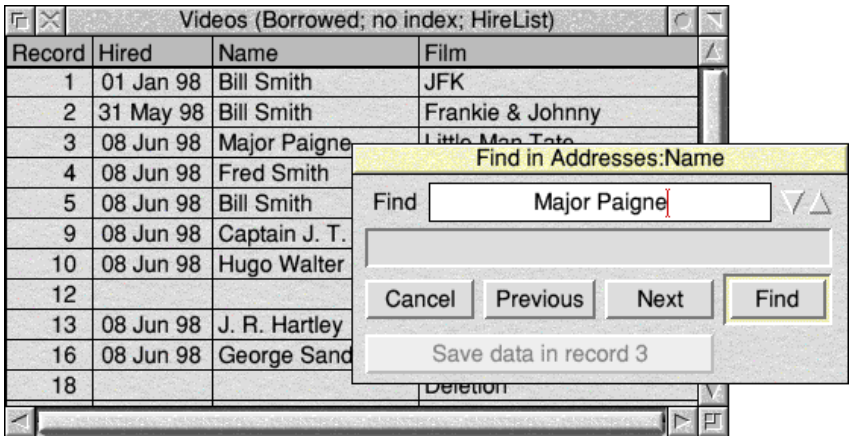
The data for most types of field may be edited direct from the table. You can thus easily update several records in turn without calling up their record cards one by one. Notes fields may not be edited in the table. Fields which cannot be edited in the record card, such as calculation fields or any field marked as non-writable, cannot be edited from the table either. Finally, table columns consisting of several fields concatenated using a field map may not be edited.

To edit a value shown in a cell, hold the ***Alt*** key and click ***Select***. For text, number, sequence, date and file fields the caret will appear in the table and you can edit the value direct. Press ***Escape*** to discard changes and ***Return*** or ***F3*** to save. You may save the value and move to an adjacent table cell by using ***Tab*** and ***Shift-Tab*** to moved right and left, and the ***Up*** and ***Down*** cursor keys to move to the previous or the next row of the table. In a multi-line text field, pressing ***Return*** will only save the changes if the caret is on the last line: otherwise a line break is inserted. Pressing ***F3*** will save the changes no matter which line the caret is on.

When you ***Alt***-click over menu, option, browser or flag field, a menu will pop up allowing you to choose the new value to be saved in the record. Again, press ***Escape*** to avoid making a change, or click elsewhere outside the menu. Because flag fields have several values which can be ticked or unticked independently, the menu for these works slightly differently. The lower part of the menu displays the possible values, and clicking on these menu entries will toggle the values but not close the menu. To close the menu click on Save to store the value or Cancel to discard the changes.

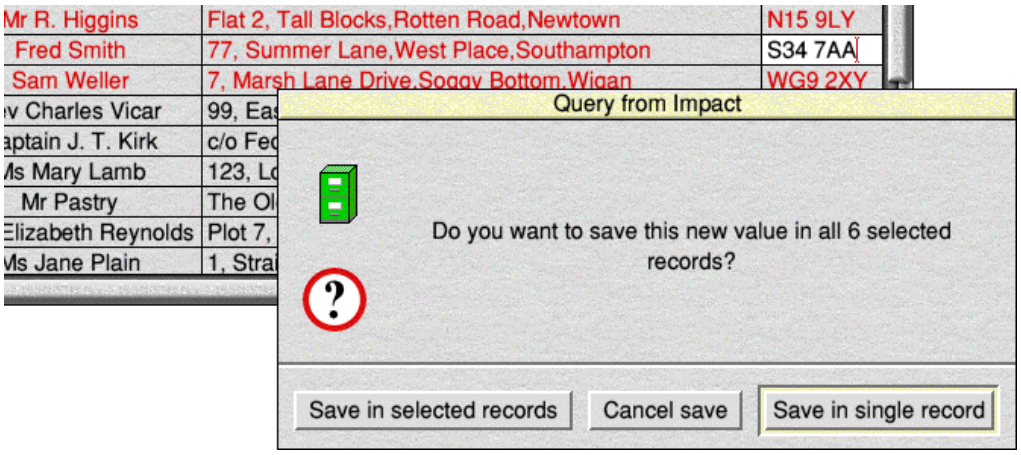


Editing a foreign key field using ***Alt***-click will bring up the Find window, allowing you to search for the record you wish to be linked. See *15.6 Entering foreign data* for further information.



9.13 Editing several selected records together

If you have two or more records selected in the table, you can save any changes to a field in all selected records at once. Find one of the selected records in the table, and simply ***Alt***-click the field to be edited to start the editing process as usual. Then when you save the change you will be prompted as to whether the other selected records should be affected too.



Please note that to avoid users accidentally editing many selected records, the option to save the change in all selected records does not appear if you move to another field or record using the ***Tab*** or cursor keys, but only if you use the ***Return*** or ***F3*** keys.

Chapter 10: Mail merging and formatted reports

This chapter describes the **merge tool**, and how to use it to send data to *Ovation Pro*, *Impression* or *ImpEmail* to create reports or perform a mail merge to send a customised letter to different people.

This facility is provided via the *Impulse* module written by Computer Concepts (now public domain). The process for each application is very similar, but any differences between using *Ovation Pro, Impression* and *ImpEmail* will be highlighted as the process is discussed. Throughout this chapter it is assumed that you are familiar with the relevant features of these applications. For the sake of brevity we will simply refer to 'the editor' for whichever application Impact is communicating with.

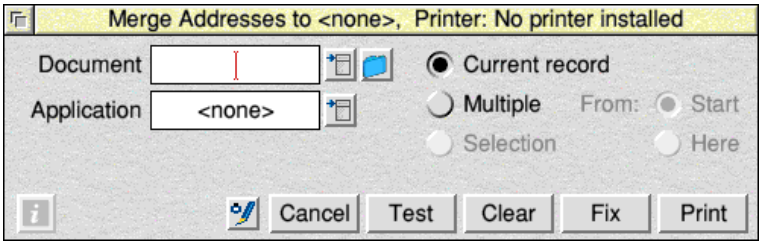
In order to merge to an application you must ensure that RISC OS has 'seen' the relevant application, or Impact will be unable to create or open the documents. For *Ovation Pro* the Impulse applet must also be active. Open the !OvnPro application directory and check the subdirectory 'Applets' to see if !Impulse is present. If not, copy it to that location from the directory called DavidPilling.Applets on your *Ovation Pro* CD. It is also available to download from David Pilling's web site.

We will start with a simple example using *ImpEmail*, before going on to describe the merge process in general and to explain the merge commands.

10.1 Introductory example

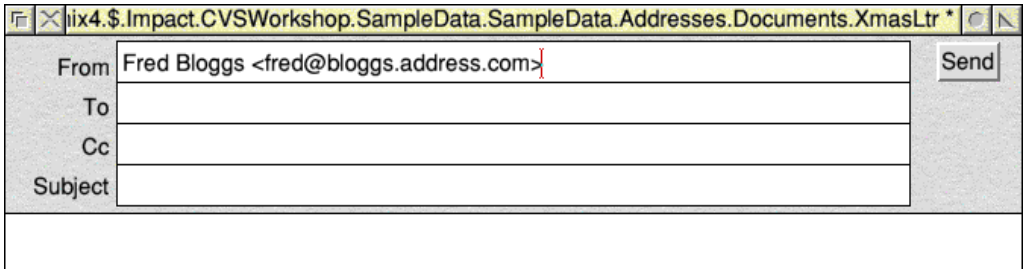
Suppose you have a database of addresses including e-mail addresses, and you wish to send an annual letter to all your Christmas card list. In the past you may have simply composed a single letter and printed out multiple copies, perhaps using Impact's **label tool** to print labels for the envelopes. Now you wish to save paper and postage costs and send the letter by e-mail. *ImpEmail* allows you to send the same e-mail to as many different people as you have e-mail addresses for, and you can personalise some of the text too.

Open the sample Addresses database provided with Impact. Click on the **merge tool** to open the **Merge window**. This window provides facilities for inserting merge commands into documents and controlling the data merge operation.



We want to create a new e-mail, so we click on the "Document" menu button and choose (New) from the top of it. This will open a window for us to choose the application we wish to merge to (*ImpEmail* will be selected if we have not previously used the merge tool with another application) and the name of the document. Type "XmasLtr" and press ***Return***. This will open a blank e-mail in *ImpEmail* for you to edit. (If you have not previously used *ImpEmail*, it will ask you for some setup information first so that it knows how to send e-mails on your computer. See *Appendix 4: ImpEmail manual* for more information.)

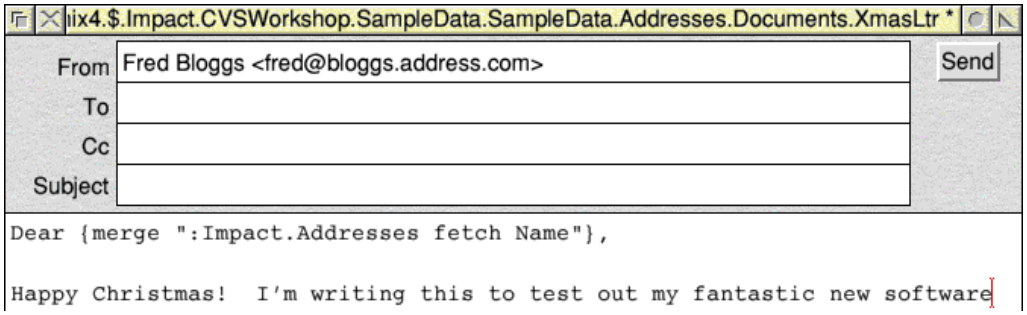
You will need to supply your e-mail address in the "From" box.



Then you need to put an address in the "To" box. As you will be sending the message to all of your card list, this is where Impact comes in. You are composing the message as if it were going to just one individual. Click on the *Edit* icon on the merge window to tell Impact that you want to insert a merge command.

Now drag the e-mail address field from the current Addresses card to the *ImpEmail* "To" box. It doesn't matter whether the current record actually has an e-mail address or is even in your Christmas Card list: all we are doing is telling *ImpEmail* that we want to insert a merge command in our document to insert the contents of that field when we perform a merge from any record.

In the body of the *ImpEmail* window you should compose your letter. You could address each e-mail personally, by starting "Dear ..." with the name of the recipient. To do this, after you have typed "Dear " you could drag the name field from the Addresses card to the *ImpEmail* window in the same way as you did for the e-mail address. Or alternatively, insert the command using the section at the bottom of the Merge window: choose "Fetch" and then select "Name" from the menu of field names. Either method produces the same result.



Now click on "Test" in the Merge window to see how your e-mail would look if you were to send it to the person in the current record.

(Of course, the letter would look better if the database had a field containing the person's name in the way you would like to address them, so that the letter read "Dear Bill" rather than "Dear Bill Smith".)

To send the e-mail to everybody on your card list who has an e-mail address, you will need to first switch on the filter "CardListEmail" which combines those requirements, and then choose "Multiple" from the Merge window. Then click on "E-mail" in the Merge window and the messages will be sent to your e-mail software's outgoing message queue. (Note, the e-mail addresses for the imaginary people on the CardList are all invalid, so if they are sent off the messages should bounce back to you. You will be able to inspect them and delete them while they remain in your queue.)

You could compose a similar letter in *Ovation Pro* or *Impression* to use for those people on your Christmas card list who do not have an e-mail address, having set up an appropriate filter.

For an example showing report generation in *Impression*/*Ovation Pro*, see *1.14 Producing reports*.

10.2 The data merge process

Merging is initiated by Impact sending a merge command to the editor, specifying the name of the document to be merged. The editor then begins to scan the document, and as it finds embedded merge commands it sends these to whichever application is specified in the command.

If these are received by Impact, and it understands the command sent, it will execute the command. This may be a command requesting data to be merged, in which case the requested data is returned to the editor. The text will be incorporated into the document, at the position of the merge command.

The command may not only request data, but may do other things, such as access records satisfying a filter, inserting a table of records, performing a calculation, etc.

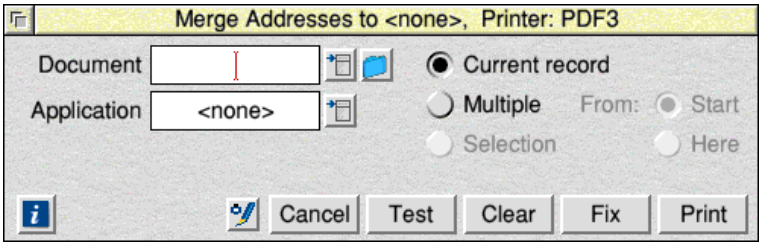
This action continues until the end of the document is reached, when the editor will tell Impact that it has finished merging.

If printing or e-mailing has been requested by the user, Impact will then send a print command to the editor, again specifying the document, and the editor will carry out the printing or e-mailing operation, and tell Impact when this has been completed.

For all this to work, the appropriate commands must be embedded in the document. As we have seen in the example these can be created easily by dragging from Impact, but they can also be created directly in the editor without having to have Impact running.

10.3 The merge window

A window similar to the following is displayed when the **merge tool** is selected. The last button will be labelled "E-mail" for communicating with *ImpEmail* and "Print" for *Impression/ Ovation Pro*.



This window provides facilities for controlling the data merge operation with *Impression/ Ovation Pro*, and provides tools to aid in creating *Impression/Ovation Pro* documents.

Clicking on the *Edit* icon extends the merge window to provide facilties for building merge commands. Click with ***Select*** to switch between the small window and the full window with command building panel, and with ***Adjust*** to switch between the small window and the *Command* box if the command building panel is taking up too much of your screen space.

10.4 Selecting a document

There are a number of different ways of selecting the document you wish to merge to.

1. Select it from the *Document* menu. This menu shows all documents stored in the selected database's Documents directory. You can click on the small directory icon to view all such files, which is the easiest way to delete any which are no longer required.
2. Save a document from your editor directly to Impact's **merge window**. This will save it in the database's Documents directory so that it is added to the *Document* menu.
3. Drag a document to the **merge window** from any filer window.

4. If the document is already loaded into your editor, type its name into the *Document* box.

Note that these last two methods will not copy the document into the database's Documents directory, but merely set up the link between Impact and the editor. If you wish to use the same document again you will need to remember where you saved it, as it will not be shown on the document menu.

If your editor is not running and has not been 'seen' by RISC OS, you will get an error when you try to open the document. You will need to open the directory containing the editor application (you can then close it again) and re-select the document.

If you close the document, Impact has no way of knowing this. You will need to re-open it using one of the above methods before the merge facilities can be used.

10.5 Creating a new document

Impact can create a new document for you for *ImpEmail*, *Impression* or *Ovation Pro*. To do this it copies a default document, which is saved in Impact's Choices directory. A default document for *ImpEmail* is supplied with Impact. You may wish to edit it to insert your e-mail address and/or your signature and replace the default tenplate.

To be able to create new a new *Impression*/*Ovation Pro* document you will first need to run the relevant editor and save a document to Impact's Choices window. Your default document may be blank, or it may contain your letterhead or other text.

See *21.12 Merge document templates* for more information on saving and editing default documents.

Alternatively, you may create a new document simply by doing so in the editor application, whence it can be saved to the Impact Merge window as described in the previous section.

10.6 Previewing the document

It is useful to be able to check that all is well before printing, and this may be done by clicking on the *Test* button.

This will cause the editor to start a merge operation, and the merged data will be displayed in the document. Only the current record will be used, even if the *Multiple* or *Selection* record option has been selected. That is, only a single document will be displayed.

Once merged, the document is not editable, and in *Impression* the pointer will change to a large cross when moved over it. The *Clear* button will restore the document to its previous state, showing the embedded merge commands, and allowing any required modifications to be made.

10.7 Permanent merging

If a permanent file copy of the previewed document is required, the embedded merge commands may be permanently replaced with data by clicking on the *Fix* button after you have clicked *Test*. This will restore the document to an editable state.

It is not possible to recover the merge commands once this has been done, as the data are now an integral part of the document, and may be freely edited as required. The document may be saved to disc for future use, or printed, e-mailed, etc., using the facilities provided by the editor. Be careful not to save this merged version over the top of the one containing merge commands, or the original will be lost.

10.8 Printing reports

The *Print* button will first cause a merge to take place, and when this process is complete, *Impression*/*Ovation Pro* will be instructed to print the merged document.

Note that the printing is controlled by *Impression*/*Ovation Pro* in accordance with the settings current in the Print dialogue box within that application, so you should check these before clicking on the merge window's *Print* button. These settings may include which pages are to be printed and in what order, and how many copies to print.

The number of copies set within the Print dialogue in *Impression*/*Ovation Pro* controls the number of identical copies of each merged document to be printed. Thus if the number of copies has been set to 2, and 10 records in a database are being merged, then a total of 20 documents will be printed.

10.9 Aborting a print run

It should be understood that mail merging is a cooperative action between Impact and *Impression*/*Ovation Pro*. If it is found necessary to abort a print run, pressing ***Escape*** while documents are being printed may have different effects depending on which application has control at the time.

However, if the ***Escape*** key is held down, then when Impact next gets control the printing operation will be completely terminated.

10.10 Sending e-mails

The *E-mail* button will first cause a merge to take place, and then *ImpEmail* will be instructed to send the message. If you are merging from multiple records, each message will be sent separately. *ImpEmail* keeps a log of which messages have been sent successfully and which generated errors.

10.11 Saving your document

Merge documents may be stored anywhere on any disc, and may be dragged into the **merge window** when required.

However, in order for them to appear on the document menu from this window, they must be stored in the *Documents* sub-directory within the database to which they belong.

This may be done by dragging the file icon from the editor's save window into the Impact merge window, avoiding the need to find and display the relevant filer window.

10.12 Selecting data records

If the document only contains merge commands specifying data fields, with no reference being made to the data record required, as may be the case with a mail-shot letter for example, then the currently selected record will be used.

If the *Multiple* option is selected from the merge window, then all records (*From start)*, or all records from the current one to the end of file *(From here)* will be merged. In this case separate copies of the report or letter will be printed or e-mailed, one for each record. Any active filters or indexes will affect which records are merged.

It is also possible to merge a selection of records chosen from a **table display** using the *Selection* option.

10.13 Entering merge commands from Impact

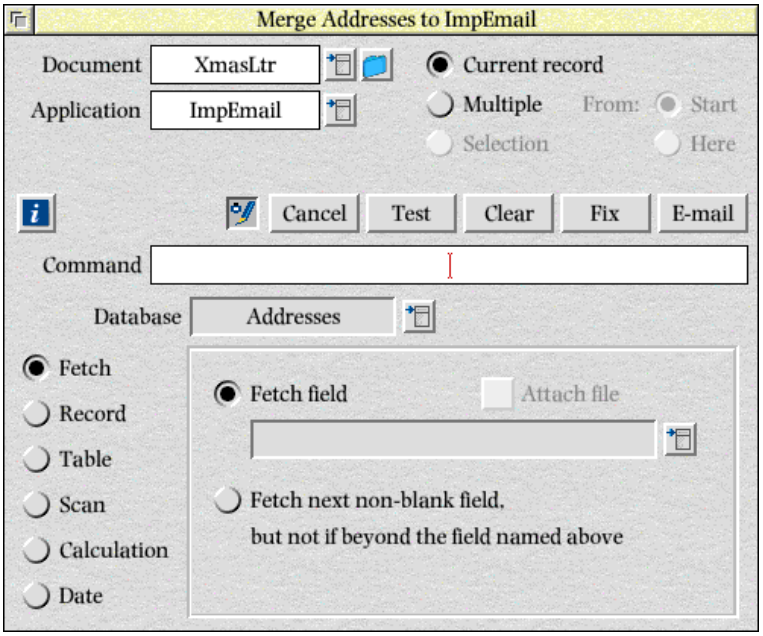
The simplest method to insert commands is to use the *Command* box at the bottom of Impact's **merge window**, which is accessed by clicking on the *Edit* icon . Then drag the *Command* box into your document at the required point.

It is possible to type merge commands directly into the *Command* box, but it is easier to construct them using the command building panel at the bottom of the merge window, as that allows all possible commands to be built without knowledge of the command syntax. The command you have constructed will appear in the *Command* box and you may then drag it to the document.

You may select any database which is on the same path as your current one, if you wish to merge data from another database into your document.

All the writable icons in the command building panel support OLE (Object Linking and Embedding) so if you need to enter a command or expression which is too long to be displayed in a box, simply ***Ctrl-***double-click to load it into your text editor. After making changes, saving the file will transfer it back to the merge window.

The following sections look at each command in turn, briefly discuss their use and explain the command syntax.



This system makes use of the *Impression* Document Description Format (DDF), or the *Ovation Pro* Document Description Language (DDL), depending on which application is in use. This means that if a merge command is dragged into another application, such as *!Edit*, what appears will be in DDF or DDL.

10.14 Technical notes

Execution context

The selection of indexes, filters, and records from merge commands embedded in documents will not have any effect on the current settings of your database card. However, when merging multiple records or a selection of records from a particular database, using the **Record next** command for that database will affect the merging process. For example, if you had nine records in your selection, and a merge document which fetched fields, moved to the next record, and fetched more fields, then the second document printed would commence with the third

record in the selection. This can be used to good effect as described in *10.25 Creating label sheets in Ovation Pro and Impression*.

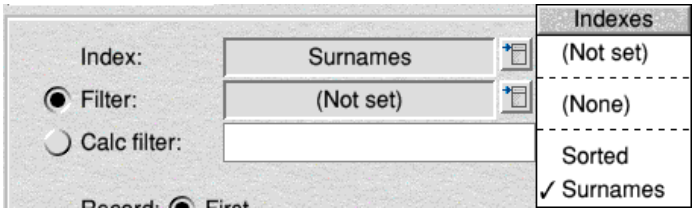
If the document merging is initiated by action script commands, then the **Record next** command will change which record is currently selected, whether or not the database is that from which the document was loaded. In addition, any selection of an index or filter in the merge commands will be carried back into the context of the action script.

Indexes and filters

The **Record**, **Table** and **Scan** commands can select indexes and filters. If no selection is made, then any index or filter which is currently active on the database will be applied.

The index and filter menus in the command building panel allow any index or filter to be selected from those defined for the database. If you wish to set a filter which is not on the database's filter menu, you can select **Calc filter** and type a calculation expression into the box to define a temporary filter for use by the merge command.

Note that any index or filter defined by a **Record** command remains in effect until turned off or replaced by another. The default choice on each menu is *(Not set)*, which retains previous settings. To turn off an index or filter, choose *(None)*.



Command syntax

It is not necessary to understand the command syntax in order to use the command building panel, but it may be useful for interpreting existing commands found in a merge document.

The syntax for each command should be interpreted as follows:

Items shown in `[square brackets]` are optional and may be omitted.

Items within `<angle brackets>` should be replaced with the appropriate name. For example, where `<field>` is given, this should be replaced with an actual field name.

Items in `(curved brackets)` separated by the `|` (bar) character, represent alternative items. Only one of the items listed should be provided.

All other items should appear exactly as given.

10.15 The Fetch command

This is the most used command, which requests data from a named field of a database to be fetched from Impact and merged into the document at the position of the merge command.

Direct insertion

With the caret positioned within the document where the merge command is to be inserted, you can insert a simple **Fetch** command simply by dragging the required field from the card into the document.

Note that this action will only result in a merge field being inserted if the *Edit* icon has been clicked so that the *Command* box is visible in the merge window. Otherwise, the textual content of the field will be inserted instead, as with a normal drag and drop action.

Using the command building panel

The **Fetch** command comes in two forms. The simple **Fetch field** fetches the contents of the field specified, and this is what you will usually want.

If the field selected is an internal or external (linked) file field, you have the option of ticking **Attach file**. This is intended for use with *ImpEmail*. The resulting command is actually a Calc command (see *Section 10.19* below) constructed to fetch the name of the file from Impact, rather than the content of the file. See *Section 10.24 Merging and attachments in ImpEmail* for more information.

The **Fetch next** command will fetch the next non-blank field after the previous field fetched, in the order listed on the menu. (This order is defined as the 'caret order' when creating a data card, or may be alphabetical, according to the setting of the **Choices** option.) In this case the field to select from the menu is the **last** field you could possibly want to fetch. You would use a succession of **Fetch next** commands to retrieve several fields, up to but not beyond the selected field. This is most likely to be useful in a database where the lines of addresses are stored as separate fields rather than one single multi-line text field.

For example, suppose there are several fields following a *Title* field, some of which may be blank, and we want all following non-blank fields up to the *Postcode* field to be used without blank lines, but do not want fields following the postcode to be printed.

In this case, we could create the merge command to fetch the *Title* field, by selecting *Fetch field*→*Title*.

Then we could create three merge commands for the next three fields using the **Fetch next** command, by selecting *Fetch next*→*Postcode*. This would ensure that if any field was blank, it would be ignored, and no fields after the *Postcode* would be used. Note that each merge command only merges a single field, and that sufficient **Fetch next** commands must be provided to complete the address.

The quantity of data which may be fetched may be limited by the application which is receiving it: there is no particular limit within Impact. It is known that *Impression Publisher* does not handle fields of 14,000 characters or more (for example, a large note field) when merging, but will handle fields of up to at least 6,000 characters.

Fetch command syntax

```
:Impact[.<database>] Fetch ( <field> | next <field> )
```

Examples:

```
:Impact.Addresses Fetch Name
```

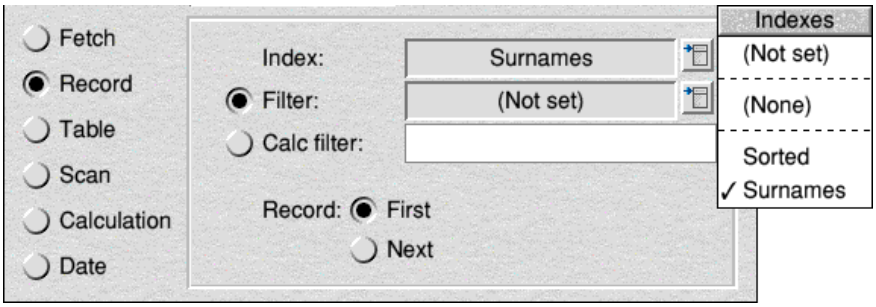
```
:Impact.Addresses Fetch next Postcode
```

The first example requests data from the named database and field.

The second example will fetch the next non-blank field after the last field fetched, provided it is not after the field *Postcode*, which will be the last field it can merge.

10.16 The Record command

This command does not directly insert data into the document, but may be used to select a particular record to be used in any following **Fetch** commands. Note that if you wish to use the current record shown on the Impact card, you need not issue a **Record** command as this record is already selected by default.



The bottom two options allow selection of the **first** and **next** form of this command, to select the first record, or the next available record after the current one, respectively.

The **next** form of this command is particularly useful for inclusion into label sheets (see *10.25 Creating label sheets in Ovation Pro and Impression,* and permits data from more than one record to be included in a single document.

The **Index** and **Filter** menus allow any index or filter to be selected from those defined for the database. Selecting an index or filter will cause the **first** command to select the first record in the index which meets the filter conditions. Similarly, the **next** command will select the next such record.

If you wish to set a filter which is not on the database's filter menu, you can select **Calc filter** and type a calculation expression into the box to define a temporary filter for use by the merge command.

Any filter or index included within a merge command will have precedence over any selected from the toolbar, and remains in effect until replaced by another or turned off with "(None)". i.e. a **Record next** command will select the next record in the previously selected index.

Record command syntax

```
:Impact[.<database>] Record <specifier> [i <index>] ([f <filter> |
c <calculation>])
```

Where `<specifier>` may be one of:

```
[next | first]
```

The specifier **next** causes the next record in the named database to be selected, while **first** causes the first record to be selected.

Examples:

```
:Impact.Addresses Record first
```

```
:Impact.Addresses Record next
```

An index, filter or both, may be specified by following the command with one or both of the following.

To include an index, add `i <index name>`

and a filter by adding `f <filter name>`

The `i` or `f` tells Impact that an index or filter name follows.

A calculation filter which is not in the database's filter list may be included by adding

```
c <calculation expression>
```

The calculation must *not* include spaces, unless they are within a quoted string. If, however, you are creating the calculation in the command building panel you may include spaces, and they will be stripped out by the command builder.

The specified index, or filter, remains in effect until replaced by another, i.e. the command

```
Record next
```

will select the next record in the previously selected index. To turn off a previously index or filter without selecting a new one, the command should include

to turn off an index: `i -`

to turn off a filter: `f -`

i.e. setting an index or filter with a minus sign instead of the name. `f -` will also turn off calculation filters set with `c`.

Examples:

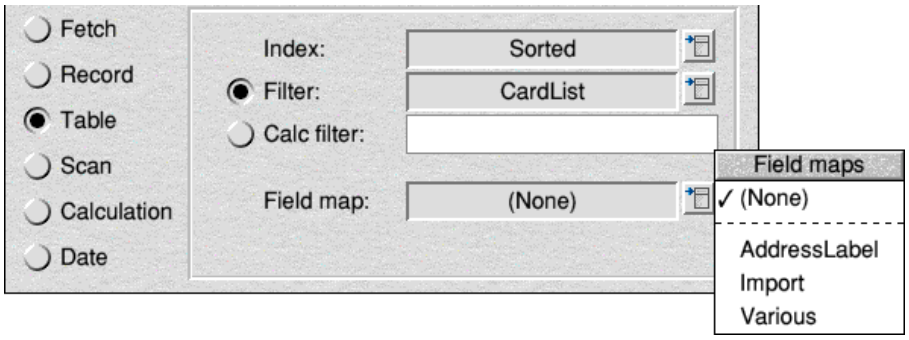
```
:Impact.Addresses Record first i Sorted

:Impact.Videos Record first c hired&&Today-hired>7
```

The first example selects the first address record in the index *Sorted*; the second selects the record for the first video which has been on hire for more than 7 days.

10.17 The Table command

This command may be used to automatically include a table of data in a document, consisting of a sequence of records. Each record will be terminated in a newline character, while all fields will be separated by tab characters (or padded with spaces in *ImpEmail*). This is modified if a **field map** is in force.

Correct formatting of the table will depend on the tab settings being correctly positioned within *Impression*/*Ovation Pro*. The merge command could also have a table style applied to it within *Impression*, defining the tab stops and other requirements. Experimentation may be required to ensure that the longest field used will fit within the specified column width.



A **field map**, **filter** and **index** may be defined for this command, by selecting them from the appropriate menus. Filters and indexes set in a Table command will not persist beyond the end of the table. However, if you do not wish to apply an index or filter to the table and one has previously been set by a **Record** command, you will need to turn it off by selecting *(None)* from the menu.

Field totals and other values are accumulated, which may be used in subsequent merge commands or calculations. (See Section 16.8, *Table functions*.)

If a **field map** is used, this may have been defined to concatenate certain fields. Tab characters will not be inserted in the output between fields which are concatenated by the field map. If you are using *Ovation Pro*, pressing ***Ctrl-I*** will reveal hidden formatting codes such as tab characters so that you can see what is going on. Another tip, if you are having trouble with your table command definition, is to set the same index, filter and field map using the tools on the record card, and then open a table window using Impact's table tool. The table in the merge document will contain the same data and columns as that opened by the table tool.

Automatic line numbering in a table in *Ovation Pro*

In *Ovation Pro* it is possible to automatically assign line numbers to each row of the table. Highlight the table merge command in the *Ovation Pro* document and select *Text➔Format* from the menu or press ***Ctrl-F***. Tick the *Auto insert* box and choose *Number* with the format you require (for example, *123/n*). Click *OK* in the *Auto bullet and number* and *Modify format* dialogue boxes. You should find that your chosen starting number or letter now appears next to the merge command. When the merge is performed, each row of the table will be numbered according to the format you have chosen.

Table command syntax

```
:Impact[.<database>] Table (all|<fieldmap>)  [i <index>] [f <
filter>] [c <calculation>]
```

Tables are exported using tab field separators, so the tab settings in *Impression*/*Ovation Pro* should be set accordingly to produce a neat layout. In *Impression* you can apply the Table style to this merge command to display a table with lines drawn between columns, etc.

Filters and indexes set in a Table command will not persist beyond the end of the table. However, if you do not wish to apply an index or filter to the table and one has previously been set by a **Record** command, or is selected on the database card, you will need to turn it off using `i -` or `f -` as described under the **Record** command.

Examples:

```
:Impact.Addresses Table all i Sorted
```

```
:Impact.Addresses Table Address f CardList

:Impact.Videos Table ReportMapl c hired&&Today-hired>7
```

The first example will include all fields in the table, and all records on the index *Sorted*.

The second example will produce a table containing the fields specified by the field map *Address*, and include all records which pass filter *CardList*. Note that the field map will specify both the field selection, and the field order.

The third example will use the calculation filter defined to produce a table of overdue videos, containing the fields specified by the field map *ReportMap1*. Note: the calculation must *not* include spaces, unless they are within a quoted string. If, however, you are creating the calculation in the command building panel you may include spaces, and they will be stripped out by the command builder. Quotes within quotes must be escaped by preceding the inner quotes with a backslash. In most circumstances it is probably easier to create a calculation filter within the database and use that!

10.18 The Scan command

This command causes the database to be scanned, subject to any active filter, as with a **Table** command, except that no data is merged. However, the field totals and other values are accumulated, and are available for subsequent merge commands or calculations. (See the example below, and *16.8 Table functions*.)

As with the **Table** command, a **field map**, **filter** and **index** may be defined for this command, by selecting them from the appropriate menus.

Scan command syntax

This command syntax is identical to that of the **Table** command.

```
  :Impact[.<database>] Scan (all|<fieldmap>)  [i <index>] [f <
filter>] [c <calculation>]
```

Examples:

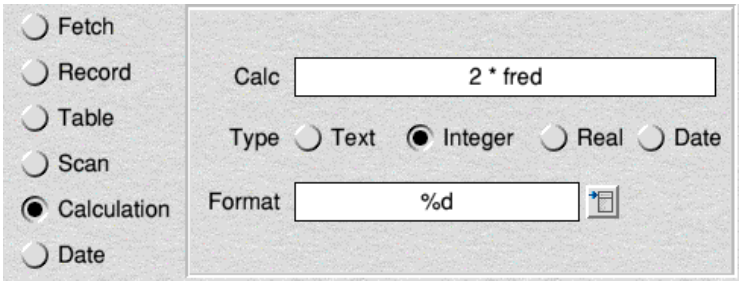
```
:Impact.Videos Scan ReportMap2 i BestSellers

:Impact.Videos Calc Total(3)/Count()
```

The first example scans the table which would be presented if you opened the **table tool** with the *ReportMap2* field map and *BestSellers* index selected.

The second example calculates the total of the values from the third column of this table (not counting the Record number column) divided by the number of rows, i.e. the average fee earned by each video which has been hired.

10.19 The Calc command

The **Calc** command allows calculations to be entered as merge commands, to be computed at merge time.



The **Type** and **Format** options allow the result type and format to be defined, in a similar manner to the facility provided when designing a card, for entering calculation fields. The format menus present popular options with descriptions and examples of the same value being formatted differently.

Calc command syntax

This command allows calculations to be directly inserted in a document, without having to have a calculation field on the record card. The syntax is

```
:Impact[.<database>] Calc <expr> [ <type> [ <format> ] ]
```

Examples:

```
:Impact.Sales Calc cost+vat

:Impact.Sales Calc 1.2*cost 1 %1.2f
```

Notes:

1. There should be no spaces within the calculation (though if you are creating it in the command building panel you may include spaces).
2. The data type codes must be one of:-
   - 0 Integer
   - 1 Real
   - 2 Date
   - 3 Text (default)
3. The format for Integer and Real should follow the *printf* syntax of the C programming language; the date format is as described in *Appendix 3: Date format codes*. Knowledge of this is not necessary when using the facilities provided by Impact, but may be of use to advanced users.

10.20 The Date command

This command causes the current date to be inserted into a document, and is useful for letter heads, etc.

The menu allows a date format to be specified. If no format is given the displayed date will be of the form 21st August 1995.

Note that this does not simply insert today's date, but inserts a merge command which will always display the current date on the day the document is printed.

Date command syntax

```
:Impact[.<database>] Date [<format>]
```

where `<format>` is any valid date format string. A format string consists of literal text characters, and special fields starting with a percent sign, indicating where specific data fields are to be inserted.

Example:

```
:Impact Date %zdy.%zmn.%ce%yr
```

If no format is given, the default format string is used. This is defined as:

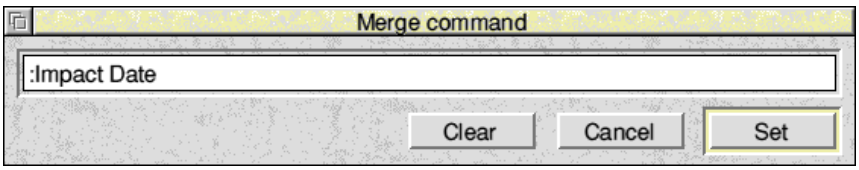
```
%dy%st %mo %ce%yr
```

which results in a date of the form: 21st August 1995

The full list of date format codes is given in *Appendix 3: Date format codes*.

10.21 Entering merge commands directly from *Impression*

To merge Impact data into an *Impression* document, a merge command must be inserted into the document at the position where the data is required. This may be done from within *Impression* using the *Impression merge command* box, available by clicking with ***Menu*** over the document window and selecting *Merge command...* from the *Misc*, or *Utilities* sub-menu.



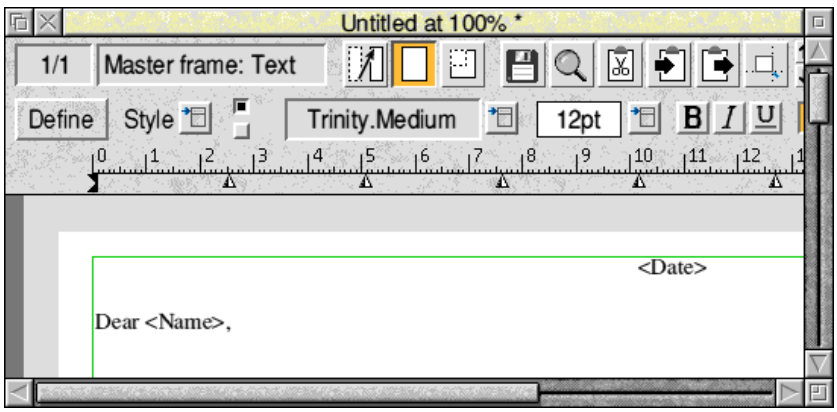
The required merge command, e.g.

```
:Impact Date
```

or

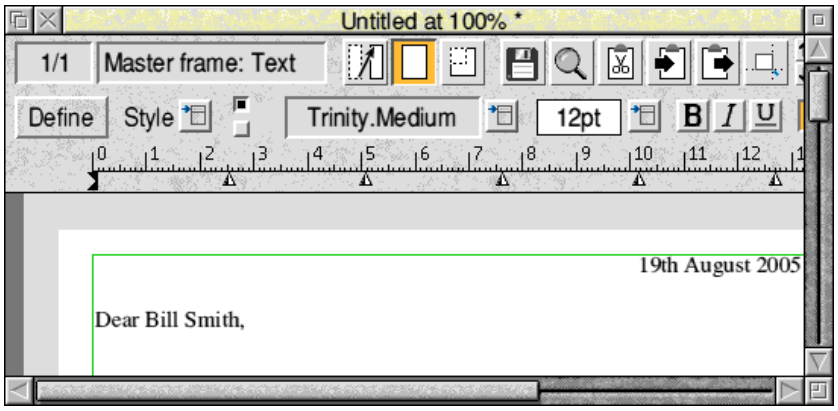
```
:Impact.Addresses Fetch Name
```

may then be entered into the writable box, and will be inserted at the cursor position when the *Set* action button is clicked. Note that since it is not *Impression* which must understand the command but the application to which is is sent, it is not possible for *Impression* to check the validity of the command.



You will see that the merge commands simply appear as <Date>, <Name> etc. (the last word of the merge command), but they must be inserted in the document using the method described above. They can be moved within the document by dragging with the mouse or using cut/copy and paste, and deleted using the ***Delete*** or ***Backspace*** key. You can use the same merge command several times in a document if required by copying.

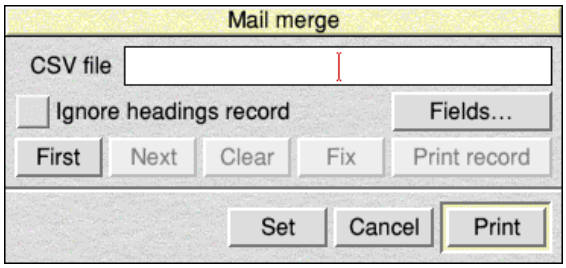
To test or perform the mail merge operation you would need to open the appropriate database in Impact, select the **Impression merge tool** and, after entering the name of the document (in this case *Untitled*) in the *Document* box, click on *Test*, *Fix* or *Print* as described earlier in this chapter.



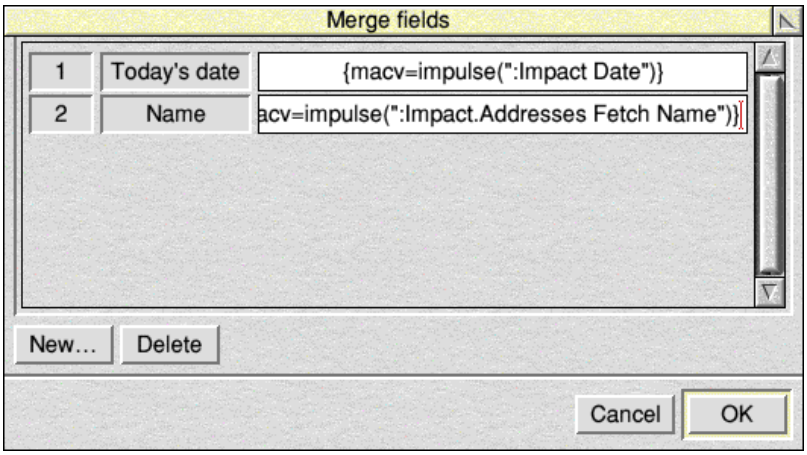
10.22 Entering merge commands directly from *Ovation Pro*

To merge Impact data into an *Ovation Pro* document, a merge command must be inserted into the document at the position where the data is required. There are two stages to this process: defining the commands and inserting them at the required positions. This may be done in either order, but it is probably easiest to define the commands first.

To define the merge command(s) required, click ***Menu*** over the *Ovation Pro* document and select *Mail merge...* from the *File* sub-menu to open the *Mail merge* dialogue box.

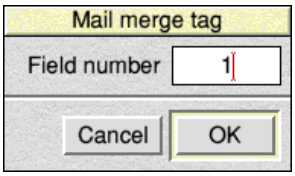


Now click on the *Fields...* button to open the *Merge fields* window. This will be blank if no merge commands have been defined for this document. To enter a new command, click on *New...* and assign it a number and name. The name can be a helpful reminder of what the command was supposed to do. The new command will be added to the list in the *Merge fields* window where you can enter the actual command you require. Note that for *Ovation Pro* the merge command is contained within a macro definition `{macv=impulse("  ")}`. Also note that since it is not *Ovation Pro* which must understand the command but the application to which is is sent, it is not possible for *Ovation Pro* to check the validity of the command.

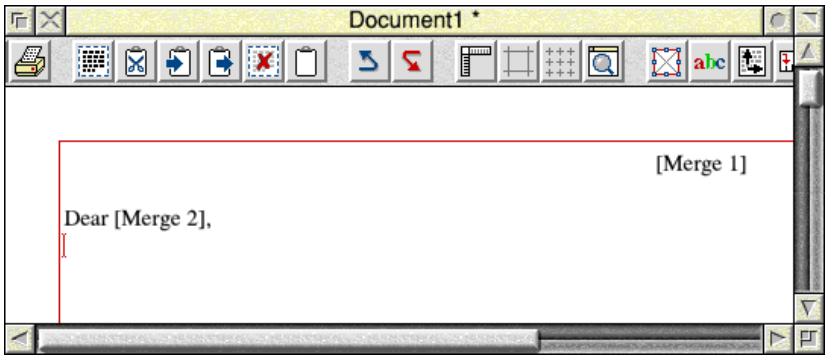


You can add or delete merge commands by clicking on *New...* or *Delete*. Click on *OK* when you have finished, and then click on *Cancel* if you wish to close the *Mail merge* window.

To insert a merge command in the document, place the caret at the required position and click with ***Menu*** over the document window. From the *Misc* sub-menu select *Insert➔Merge tag* (or press ***Ctrl-Shift-M***).

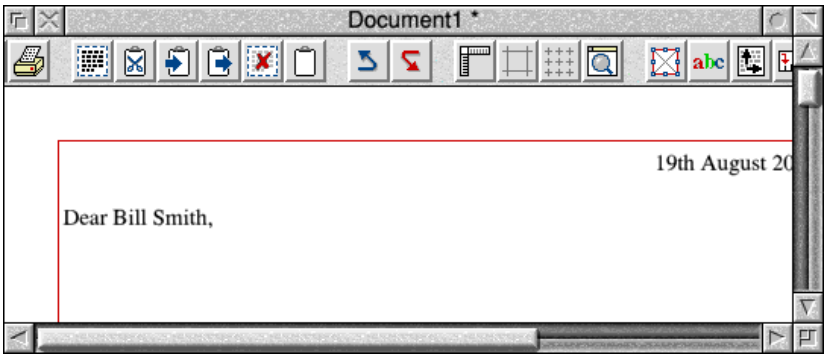


The *Mail merge tag* window will open in which you simply type the number of the merge command you defined earlier.



You will see that the merge commands simply appear as [Merge 1], [Merge 2] etc., but they must be inserted in the document using the method described above. They can be moved within the document by dragging with the mouse or using cut/copy and paste, and deleted using the ***Delete*** or ***Backspace*** key. You can use the same merge command several times in a document if required, and they do not have to appear in numerical order.

To test or perform the mail merge operation you would need to open the appropriate database in Impact, select the **Ovation Pro merge tool** and, after entering the name of the document (in this case *Document1*) in the *Document* box, click on *Test*, *Fix* or *Print* as described earlier in this chapter.



10.23 Entering merge commands directly into *ImpEmail*

To merge Impact data into an *ImpEmail* message, a merge command must be inserted into the message at the position where the data is required. This is simply done by typing:

```
{merge ""}
```

with the merge command inserted between the double quotes.

For example:

```
{merge ":Impact Date"}
{merge ":Impact.Addresses Fetch Name"}
```

If you need to include double quotes in the merge command, they should be doubled up:

```
{merge ":Impact.Addresses Table AddressLst c Title==""Mr"""}
```

This corresponds to a calculation filter with expression

```
Title=="Mr"
```

If you wish to include braces in your message for other purposes, an opening brace should be escaped by preceding them with a backslash: \{, and similarly a backslash must be escaped with another backslash: \\

10.24 Merging and attachments in *ImpEmail*

In *ImpEmail* an attachment is recorded simply as a link to a file, the content of which will be read and attached to the message when the e-mail is sent. This makes it perfect for data merging, with the file being fetched from one of Impact's internal or external file fields, or from a file whose name is worked out on the basis of some other aspect of the data.

The merge command required is a little more complicated than a plain **Fetch** command, because we need to ensure that the filename of the file is fetched from Impact, not the content of the file, but Impact tries to make it easy to add merged attachments to *ImpEmail*.

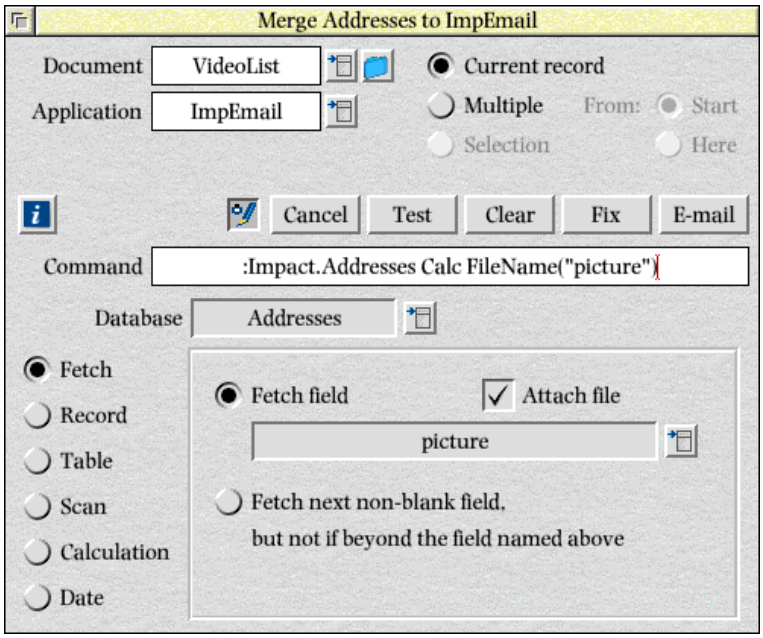
The simplest way is by dragging the field from the Impact record card and dropping it on the *ImpEmail* attachments area. In this example, the user has dragged the picture field from the sample Addresses database to the *ImpEmail* editor attachment area:

The file icon appears as a little cog wheel, to indicate that the merging machinery will work out which file is to be attached at the point of sending. For an **internal file** the name of the attachment will be set up the same as the name of the field, because the filenames within Impact's database structure are not meaningful outside Impact. But an **external linked file** field will show as "[unnamed]" and the actual filename will be substituted on merging. In each

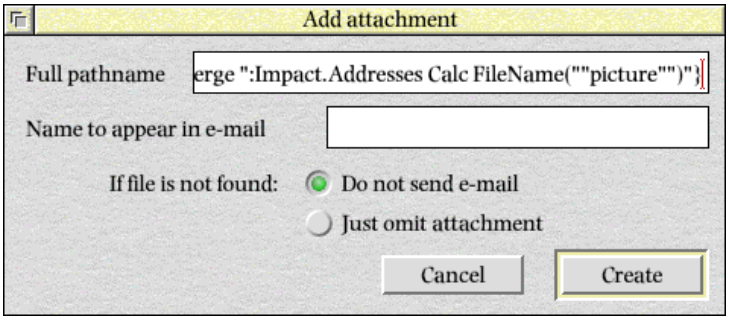
case a suitable file extension may be added to the end of the filename to help Windows PC users. You can override the default naming behaviour using the attachment attributes window which can be accessed by double-clicking the attachment in the *ImpEmail* editor.

The simple drag and drop method will not work in all circumstances. It is not possible to drag the file field if it is shown as a button, or if the field is empty for the current record.

The command builder, found in Impact's **merge window** offers a straightforward solution for all file fields by means of an **Attach file** tick box as part of the **Fetch** command. This actually constructs a **Calc** command to fetch the filename, rather than the content of the file, as this is what *ImpEmail* needs.



To add the command to *ImpEmail*, first select **New attachment** from menu over the attachments area in *ImpEmail* and drag the command from Impact's **merge window** and drop it in the **Full pathname** box of the form:



If you want to specify the name which should appear to the recipient of the e-mail, fill in the second box of the form also. This box can also include merge commands.

The **Calc** command which is used to fetch the filename is fairly straightforward, though once the double-quotes are doubled up for use in *ImpEmail* it looks rather worse:

```
{merge ":Impact.Addresses Calc FileName(""picture"")"}
```

ImpEmail offers you a choice as to whether the e-mail should be sent or not when there is no available file to attach from a particular record.

10.25 Creating label sheets in *Ovation Pro* and *Impression*

Impression/*Ovation Pro* may be used to create labels, in addition to the facilities built into Impact, and this may be useful where a more complex label design is required.

To create a document to be used for printing labels, it is only necessary to create a single page, which if merged using the **Multiple** or **Selection** option will be used several times to go through the whole database or records selected from a table.

Create the first frame

The easiest way to proceed is to create a single frame in an *Impression*/*Ovation Pro* document, the same size as the individual labels on your label stationery, and insert the required merge commands. At this stage it is a good idea to check that the font style is set to a suitable type and size for the label. Homerton medium is an easy style for automated sorting machines to read.

Select the record required

The first command required will be the **Record next** command, to ensure that each label uses data from the next available record. Note: you will need to delete this command from the first label on the sheet when the document is complete, to ensure that the first label uses the first record in the database (or the record currently displayed in the card). The *Multiple* option will move the record number on for the start of subsequent pages.

Merge the first field

Select the **Fetch** command specifying the first field to be used, probably a name field. This merge command may be on the same line as the previous record command, which will not itself cause data to be merged, and so will not take up space.

When placing merge commands by dragging them into the document, the caret will remain at the beginning of the <Merge> insert. It is therefore necessary to move it to the start of the next line ready for the next field, otherwise the next <merge> will be placed before the previous one, rather than after it.

Subsequent fields

The following fields may be merged explicitly in the same way as the first, or by using a series of **Fetch next** commands, specifying the last field to be used on the label, probably a postcode field. Providing you supply enough **Fetch next** commands this will fetch all non-blank fields after the name field, finishing with the postcode field.

The order in which fields are merged when using the **Fetch next** command is that shown on the field menu. This order is defined as the 'caret order' when creating a data card, or may be alphabetical, according to the setting of the **Choices** option (see *21.5 Field menus*).

Check the single frame first

When this single label is finished, it can be checked by using the *Test* option from the Impact **merge window**. If your label document has not been saved, then type the name at the top of the *Impression*/*Ovation Pro* window into the *Document* box in this window, so that Impact can tell *Impression*/*Ovation Pro* which document is to be merged.

Replicate the frame

When all is well, select the label frame by clicking in it, and select *Edit→Copy frame* from the *Impression* document menu or *Edit→Copy Text frame* from the *Ovation Pro* document menu (or the ***Ctrl-C*** keyboard short-cut), to place a copy of the label onto the clip board.

It is now only necessary to duplicate the label using the *Edit➔Paste frame* or *Edit➔Paste Text frame* menu option, or the ***Ctrl-V*** keyboard short-cut. Labels will be filled during the merge operation in the order in which they are created, so it may be helpful to duplicate them across and down the page.

The exact placement of frames should exactly match the position of labels on the stationery, which may be done precisely with the aid of a ruler, and the *Alter frame* dialogue box in *Impression* or *Modify text frame* dialogue box in *Ovation Pro*.

Finally, do not forget to remove the initial **Record next** command from the first label on the sheet, or one record will be skipped on each printed sheet!

Printing the labels

If you wish to use a particular selection of records you will need to set these up within Impact in a table, filter or index and have that active. If you wish to use all records, ensure any such filter or index is not active.

To test or print all the required records you should choose *Multiple* (or *Selection*, if you wish to use a selection chosen from a table) from Impact's merge window, and click on *Test* or *Print*. Note that *Test* will only produce one page of labels, but *Print* will go through the entire database or selection, producing as many pages as required.

Chapter 11: Printing labels

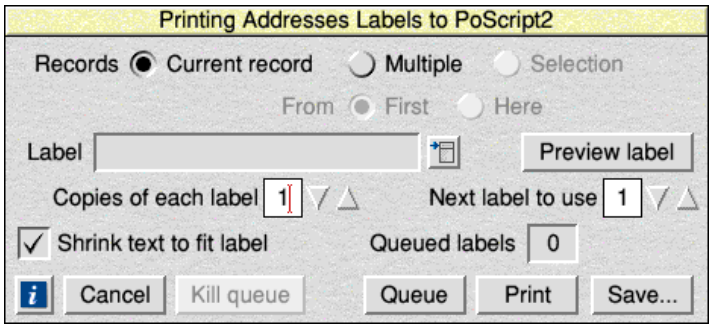
This chapter describes how to design and print labels using the **label tool**. Labels may also be printed using other label design software such as *LaBella*, by exporting the Impact data as a CSV or TSV file (as explained in Chapter 18: *Exporting data*) and loading it into that application. Printing labels using *Ovation Pro* and *Impression* is discussed in Section 10.24, *Creating label sheets in Ovation Pro and Impression*.

11.1 The LabPrint application

This application supplied with Impact handles all label printing operations using the label tool. This is done using inter-application messages, and is completely controlled from within Impact.

Label sheet design is carried out using *LabPrint*, and this is described later.

11.2 The label window



This is displayed by clicking on the **label tool** from a database toolbar.

This window allows the selection of pre-defined label styles, and controls the label printing process. Labels may be printed directly, saved as a file for printing later, or queued for later printing in a batch.

As with other Impact tools, the records used to print labels are controlled by any currently active **index** or **filter**.

When finished, the label window may be closed by clicking the *Cancel* button, or by clicking on the **label tool** icon to cancel it.

Label styles

To print a set of labels, it is first necessary to select the label style required, and this may be done using the label selector near the middle of this window. Simply click on the *Label* menu icon, and select the label required from those presented.

When you do this, it is *LabPrint* which displays the menu, and there may be a few moments' delay while this application is started, if it is not already running. The chosen label style will be shown in the label selector display. Click on *Preview label* to see how data from the current record would look in this label style.

The label definitions are held within *LabPrint*, which is supplied with a set of pre-defined label styles from the Avery Laser Label set. You can design your own label styles, or modify those supplied, from within *LabPrint*, as described later.

Multiple records

As first displayed, the *Current record* option is selected. In this mode the record currently displayed in the card will be printed. If you select *Multiple*, you can choose to print all records (*From first*) or all records starting from the currently displayed record (*From here*). Or you can

print a selection of records by opening a **table**, selecting those you want and clicking on the *Selection* option.

Copies of each label

The number of copies of each label may be set using the adjuster arrows or by typing in the required number. This sets the number of duplicate labels to be printed from each record.

Next label to use

Very often you may wish to print on a partly used sheet of labels, and this is possible by setting the first label to be used, either using the adjuster arrows or by typing the required number into the box. In this context, labels are assumed to be numbered across and down the sheet.

Caution! Before inserting a partly used sheet of labels into your printer, please consult your printer manual and the label packet to check whether this is advisable.

Shrink text to fit label

Selecting this option will mean that if the text to be printed on a label is too large to fit at the predefined point size (e.g. one person's address is unusually long), the text for that particular label will be reduced to a size which will fit. If this option is not ticked, the text will be truncated so that it fits onto the label, so you may not get the complete address.

11.3 Field maps

You will probably also need to choose a **field map** to specify which fields in each record are to be used and in which order. You will be warned if no field map has been selected when opening the label window.

Any blank or empty fields will be skipped during label printing. If you specifically want blank lines, enter a space character into the appropriate field.

11.4 Printing labels

This is done by clicking on the *Print* button. The labels will be immediately printed by the currently selected printer driver.

If there are queued labels (see below) the *Print* button will read *Print Q*, and the queued labels will be printed instead of the currently selected ones.

11.5 Saving labels

A file of labels may be saved to disc for later use by clicking on *Save*, which will cause a standard *Save as* box to be displayed. The label icon may then be dragged to a filer window, or directly to a printer on the iconbar. A saved label file may be printed at a later date if required, either by dragging it to a printer, or by double clicking on it.

Note that the file contains all required label texts, as extracted from the database when the file is saved. If the database is subsequently modified, these changes will not be reflected on the labels.

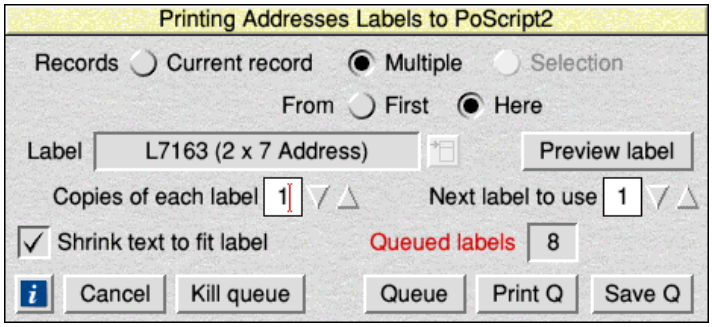
Other choices, such as the name of the label style, the number of copies, and the next label to be used, etc., are stored within the label file, and will be used at the time of printing. Note that label style name is saved, so that if this label definition is subsequently modified, then the modified definition will be used at print time.

Label printing is handled entirely by *LabPrint*, so that Impact does not need to be installed to

print a label file. If the computer has 'seen' *LabPrint*, then double clicking on a label file, or dragging the file to a printer driver, will result in the labels being printed automatically.

11.6 Queuing labels

If you click on the *Queue* button, the label will be appended to a file called *LabelQueue* stored within the Impact application. Once this has been done, you will not be able to change the label style until the queue has been printed, as all labels in the queue must be printed on the same style sheet(s).



While the display shows that there are queued labels, clicking on the *Print Q* or *Save Q* buttons will cause the queued labels to be processed.

When this is done, the settings on the label window will be obeyed, so that if you had had the number of copies set to two when queuing the label, and still have it set to two when printing the queue, you will get four copies printed.

This also means that you can set the first label to be used (which will have been ignored during queuing), to take account of partially used sheets, before clicking the *Print Q* button.

You can abandon the current queue by clicking on the *Kill queue* button.

11.7 Choosing a default label style

You may find that whenever you print labels you use the same label style each time. To avoid having to choose this style every time you use the label tool, you can set a default label style in the *LabPrint* application. *LabPrint* will be installed on the iconbar as soon as you select the **label tool** from the Impact toolbar. Otherwise it may be started in the normal manner by double-clicking on it in its Filer window.

To choose the default label style, click with ***Menu*** over the LabPrint iconbar icon and select *Choices...* from the menu. In the dialogue box presented, tick *Default label style* and choose the style you require from the menu. You will need to click on *Set & save* to save the choice you have made for future use.

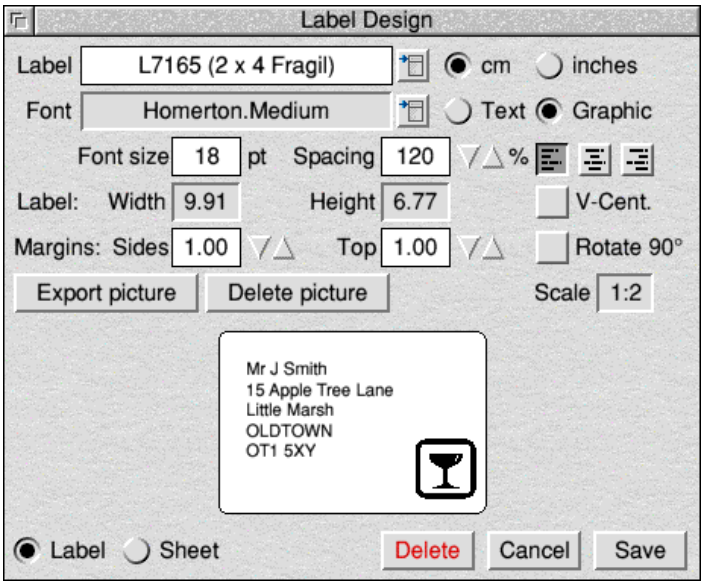
Even if you have a default label style set, it is still possible to choose a different style when required, simply by selecting a different label style from the menu in the Label window.

11.8 Designing labels

This is done from within *LabPrint*, which should be installed on the iconbar. This is done automatically if you select the **label tool** from the Impact toolbar. Otherwise it may be started in the normal manner by double-clicking on it in its Filer window.

11.9 The label design window

To display the Label Design window, select *Create...* from the *LabPrint* iconbar menu.



11.10 Selecting a label

To select a label, click the on the *Label* menu icon, and choose from the styles displayed. The name of the selected style will be displayed, and an image representing the style will be shown in the Label Design window.

If the label size is too large for the allocated display area, it will be scaled to half size or smaller, the scale used being displayed. At 1:1 the size of the displayed label will be close to actual size, although variations in screen sizes and modes will cause some degree of error.

The Label Design window has two modes, *Label* and *Sheet*, selected using the two option buttons in the lower left of the window. To display the single label design settings, ensure that the *Label* option is selected. To define the dimensions and spacing of labels on a sheet, choose *Sheet*.

11.11 Types of label

There are two basic types of label, **text** and **graphic**, chosen by selecting the appropriate option button in the top right of the window.

Text labels

These are printed by sending individual character codes (ASCII) to the printer, these being rendered in the printer's built-in font. You are only likely to want to use these with a dot matrix or daisy wheel printer, as you have no choice of font and cannot include any drawings.

Graphic labels

These are printed by sending a picture of the label to the printer as a bitmap-image, and has the advantage of being able to use any font available on the computer, and to include pictures or diagrams on the label.

11.12 Selecting a font

This applies to **graphic labels** only. To select a font, click on the *Font* menu icon, and select from the fonts displayed. The name of the selected font will be shown, and the label display will show an address in the selected font.

The *font size* should be selected by entering the required point size in the box provided.

To help with automated mail sorting by the postal service it is recommended you use a clear font such as Corpus or Homerton, and avoid fancy script.

Line spacing is defined as a percentage of the font point size, so that 100% represents single line spacing. However, it is normal to allow a little more than this, 120% being typical. You can type in a value, or use the arrow icons in the normal way.

11.13 Positioning the text

The exact position of the label text may be adjusted using the *Margins* controls and the justification buttons. As adjustments are made, the displayed text will be moved accordingly. The size of the label itself, which is set on the *Sheet* design pane, is shown above the margin settings for your information.

11.14 Using images on labels

To include images on labels (**graphic labels** only), such as the glass in the example shown, they must first be created as Draw files and then dropped onto the *LabPrint* Label Design window. To create an image which will be positioned correctly, first note the size of the label you are going to use. In *!Draw* you may find it useful to turn on the display of the grid. Mentally position the label in the bottom left corner of the *!Draw* window, and use the grid to position the image in the right place on this label. Then save the draw file to the *LabPrint* Label Design window to try it out. You can adjust the position of the text on the label by altering the margin settings.

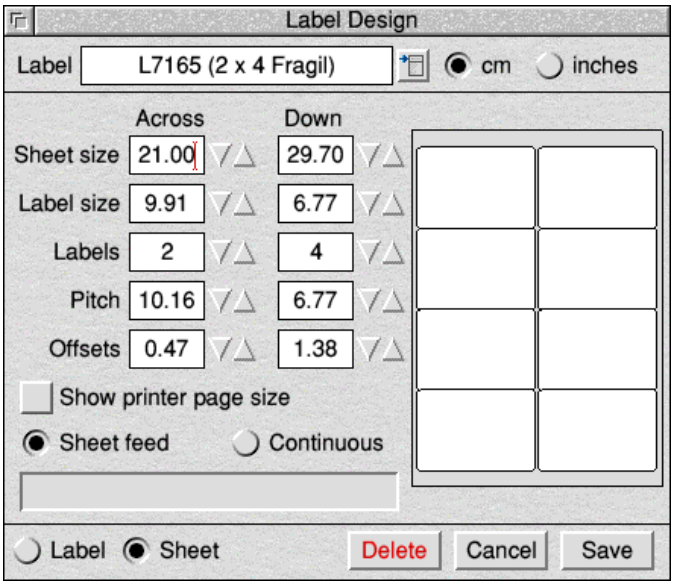
Existing images on label designs may be edited or moved by clicking on *Export picture* to load them into *!Draw*. Images may be removed from label designs by clicking on *Delete picture*.

11.15 Rotating text and images

You may wish to rotate the content of the label by 90°, for example if you wish to print in landscape format directly onto envelopes which are too big to go in your printer the other way round, or if you simply wish to print sideways on a sheet of labels. To do this just tick the *Rotate 90°* option, having set up the label design as if for portrait mode printing. (The sheet design part of the Label Design window will still show the labels in portrait mode, which is the way the sheet should be inserted into the printer.)

11.16 Label sheet design

Selecting the *Sheet* option will cause the centre panel of the Label Design window to display the label sheet design values. This displays a scaled representation of the label sheet, showing the selected layout of labels. It also shows a warning in the grey box if the numbers you have put in make the labels too large for the sheet or if the labels will overlap each other.



Choice of units

The following adjustments are concerned with actual dimensions, and these may be displayed in either inches or centimetres, whichever is the most convenient.

When altering sizes you can use the adjuster arrows to change the size in steps of 0.1, or by holding down the ***Shift*** key, in steps of 0.01. Using the ***Adjust*** mouse button reverses the arrow directions, in the standard manner.

For large adjustments it may be quicker to type in the required dimensions.

Sheet size

This is set by default to the size of an A4 sheet, but may be adjusted for other sizes.

If you select *Show printer page size*, the size of paper defined for the current printer driver will be automatically displayed. In this mode the print margins will also be displayed in grey, as a rough guide. This may indicate that parts of labels near the sheet edges may not be printable.

In this case it may be worth checking that the printer's margins are set correctly, and really do reflect the capability of the printer. If this is correct, there is unfortunately little one can do, except change printers, or more likely, label sheets.

Label size

This is the size of each individual label on the sheet.

Labels

The *Labels* boxes control the number of labels in each direction on the sheet. If you cannot get the required number of labels to fit, check the individual label size.

Label pitch

The *pitch* measurement refers to the distance from one edge of a label to the corresponding edge of the next, not to the space between labels. This measurement should be taken across several labels, and the result divided by the number of labels. This will avoid minor errors accumulating across the sheet.

Some types of label, often known as die-cut labels, have a slight gap between each label, and rounded corners. For these labels the pitch values should be larger than the label size settings. Other labels, sometimes called butt-cut, have no gaps between the labels, and square corners. For these the label size and pitch values should be the same.

Label offsets

These dimensions provide for margins at the top and left edges of the sheet.

When using continuous fan-fold paper in dot matrix printers, you should set the top margin to zero, and align the paper with the first label. The setting of the left margin should take into account the horizontal paper position in the printer.

Paper type

The *Sheet feed* or *Continuous* options simply control the sending of form feed control codes to the printer while printing **text type** labels. *Continuous* should be selected when using continuous fan fold paper, as you will not want the entire sheet to be ejected after one or two labels have been printed. With sheet fed printers, such as laser types, a form feed control code will required to tell the printer to eject the page, so *Sheet feed* should be selected.

Note that these settings are not required for **graphic** label printing.

11.17 Saving the design

To save a new design, enter a suitable name in the writable *Label* icon at the top of the dialogue box, and click *Save*. You can, of course, overwrite an existing design by using the same name.

To abandon any changes made, click on the *Cancel* button.

You can delete a saved label using the *Delete* button.

11.18 Using a different sample text in LabPrint

To use a sample text of your choice when designing labels, simply drag a CSV or text file containing the text you wish to use to the LabPrint label design window. The new text will be displayed on all designs until you quit LabPrint.

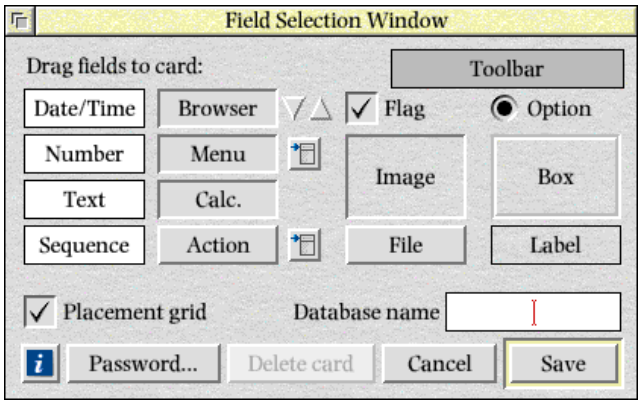
Chapter 12: Creating a database

This chapter describes the process of designing a new database, and gives an example Name and Address database. The subject of **relational databases**, and how to include **foreign fields** in a card, is fully described in Chapter 15: *Relational databases*.

12.1 A simple example

What follows is a description of how to create a simple Name & Address database. All the actions described are more fully explained in a later chapter, so do not worry about the details for the moment.

First, select the *Create...* option from the iconbar menu. When you do this a blank (untitled) card will be displayed with a toolbar containing a default set of tools, together with the **Field selection window** shown below.



This window contains examples of all the field types available. Each field type is described in more detail in Chapter 13: *Field types*.

12.2 Placing and editing labels

First place a field label by dragging the *Label* box from the **Field selection window** into the top left corner of the blank card (the black outline box will not be copied). This is not a database field, but a simple text item used, in this case, to label the real field to be placed next to it.

To change the text into something more useful, click ***Select*** once over the text *Label* you have just placed on the card. When you do this, the text will change into a fully editable icon, with the caret placed in it ready for editing. Delete the word *Label* and insert *Name*. When you press the ***Return*** key, the edit icon will change back, leaving you with the new text.

12.3 Placing and editing a text field

Next, place a text field by dragging the white *Text* field from the **Field selection window** and place it to the right of the word *Name*. If *Placement grid* is ticked, all placed items are snapped onto an invisible grid to aid alignment, but do not worry too much about exact positioning at this time.

This text field, as placed, is too short, but you will find you can drag the right hand edge out longer using the ***Adjust*** button. Drag it out to a couple of inches wide.

If you now double-click ***Select*** over the field, the **Edit text field** window will be displayed, allowing various field attributes to be set. The only thing we need to do is to set the *Max text length* longer than that shown. Use the arrow icons to bump the number up to 20. Note that this has no effect on the displayed width of the field, but sets the amount of disc space available to

store the text entered.

12.4 Multi-line text fields

Now repeat the above procedure and create a second text field immediately below the first, but change the label to read *Address*.

We need more than one line for the address, so double-click ***Select*** over the lower text field to display the Edit text field window again, and increase the number of *Text lines* to three using the adjuster arrows. We also need more than the displayed maximum text length, so click the caret into the writable icon and edit the number to read 40 (or use the arrow icons as before).

12.5 Adjusting the card

Clearly the card itself is too large, and we do not really need the scroll bars. First, use the window size icon (the lower-right corner) and adjust the window to a suitable size.

The scroll bars can be removed by clicking with ***Menu*** on the card and following the arrow to the right of *Card*, which will cause the **Card attributes window** to be displayed. Do this, and click all options off.

12.6 Saving the database

The only thing left to do now is to give the database a name, and save it to disc. To do this, click the caret into the *Database name* field in the **Field selection window**, enter *Example*, and either type Return, or click the *Save* button.

When you do this, the **Field selection window** will close, and the card is ready for you to enter data, or close the database as required. You can subsequently re-open the database from the iconbar menu.

12.7 Editing a card design

An existing database card may be edited at any time, even when there is data in the database. To do this, click with ***Menu*** over the card and choose *Edit card*. This will open the **Field selection window** so that additional fields may be added, and allow existing fields to be deleted or edited.

12.8 A choice of field types

All fields, of whatever type, are placed and edited in the same manner as described above. See *Chapter 14: Changing field attributes* for a fuller description.

When designing a database, think carefully about what information you need to store, and what form it should take. It is a good idea to make a rough sketch of your proposed card before you start, and think about how you propose to use the data.

For example, if you are designing an address database for mail merging, you may find you need a field containing a suitable familiar name, such as Sue, for use in the letter salutation ‘Dear Sue’, in addition to a more formal name for the address label. You may also need fields containing Mr/Mrs/Ms etc., full forenames, and initials.

But do not worry if you get it wrong the first time, as Impact lets you add fields, or modify existing ones, even after you have started to enter data.

All field types are described in *Chapter 13: Field types*.

12.9 Multiple cards

Where a database needs to hold many fields, it may be appropriate to design two or more cards, and split the fields between them in some logical manner.

To create a second card, click ***Menu*** over the first card and choose *New card* from the menu.

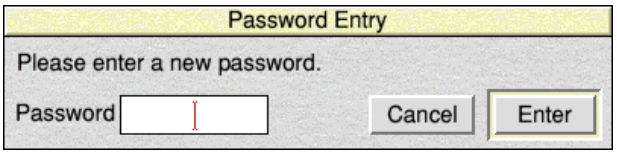
If you want to add a duplicate field to a card, simply drag the required field from the first card into the card being edited.

Data from duplicate fields is only stored on disc once, but is displayed on all cards containing the duplicates.

12.10 Setting a password

One reason for having multiple cards may be to keep sensitive information together on one card, where it may be protected with a password.

Clicking on the *Password...* button in the Field Selection window will display the password entry box shown below.



To provide password protection to the card, simply enter a suitable password into the writable icon, and click on *Enter*. As you type, the password will not be shown in case this is being observed. You will be asked to enter the password twice for verification before it is accepted.

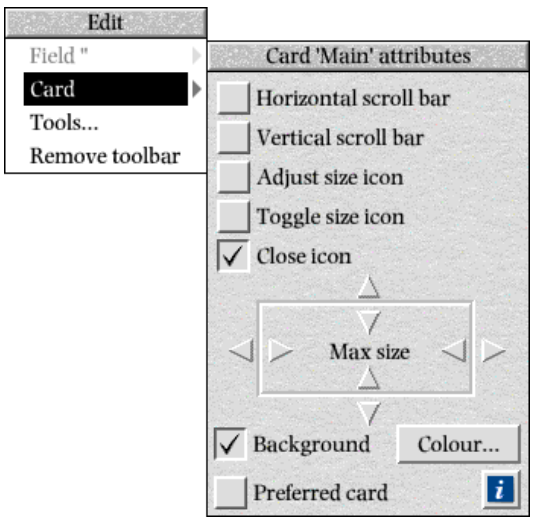
To remove a password, clear the entry field and click on *Enter*.

DO NOT FORGET YOUR PASSWORD!

Please note that setting a password on a card does not provide a high level of security. The protected data is not encrypted in any way and thus is accessible to anyone prepared to scan through the database files directly.

12.11 Card attributes window

Clicking with ***Menu*** over the card and following the arrow to the right of *Card* causes the **Card attributes window** to be displayed.



This box allows the card size to be adjusted, and the various scroll bars and window icons to be

selected as required. By default, scroll bars are provided, but these will generally be removed after the card has been correctly sized to suit the contained fields. Note that even if you choose not to include a close icon it will be possible for the user to close a card from the **card menu**.

The maximum size

The arrow icons may be used to set the card's maximum size. Note that this is not the displayed size, and the effect of these buttons may not be seen directly unless scroll bars are visible, or the card has been toggled to full size.

A card may be arranged to toggle between two set sizes, and this is useful where some data fields will not often be needed.

To do this, first use the adjustment arrows in the **Card attributes window** to set the maximum card size to be large enough to hold all the fields. Then reduce the card's displayed size by dragging the size icon in its lower-right corner until the unwanted fields are no longer visible. For convenience they should be positioned at the bottom of the card.

Then remove both scroll bars, but leave the toggle size icon. Save the database while the card is in the small state. It may then be toggled to the larger size when in use, revealing the hidden fields as required.

The card background

Ticking *Background* enables you to choose a background colour for the card.

The preferred card

Tick *Preferred card* to make this card the one which is opened when the database is opened.

12.12 Copying fields

Single fields or a selection of fields may be copied by choosing *Copy* from the *Field* sub-menu opened from the first item on the **card edit menu**. This enables you to easily create new fields with the same or similar characteristics to existing ones. They may also be copied by certain drag and drop actions as described in the table below, depending on whether you are dragging to the same card, another card on the same database, or a card on another database.

Mouse action	Same card	Same database	Another database
Drag	Move	Copy	Copy as foreign
Shift-drag	Copy as new	Copy as new	Copy as new
Ctrl-drag	Copy as foreign	Copy as foreign	Copy as foreign

Copy as new creates new fields in the same way as choosing *Copy* from the *Field* sub-menu. It creates a completely new field based on the characteristics of the old one. The name of the new field will be blank. Action scripts will not be copied as the field does not yet have a name.

Copy allows a field to appear on more than one card of the same database. Both copies will be updated in tandem.

Copy as foreign creates a foreign version of the same field. It is possible to create foreign references to the same database in this way.

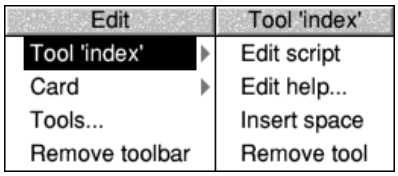
With all of these drag methods it is possible to copy a selection of fields all at once.

12.13 Editing the toolbar

To add or remove a toolbar, or to change the tools which are shown on the toolbar, you need to be in card design mode. (For an existing database, select *Edit card* from the menu over the card.)

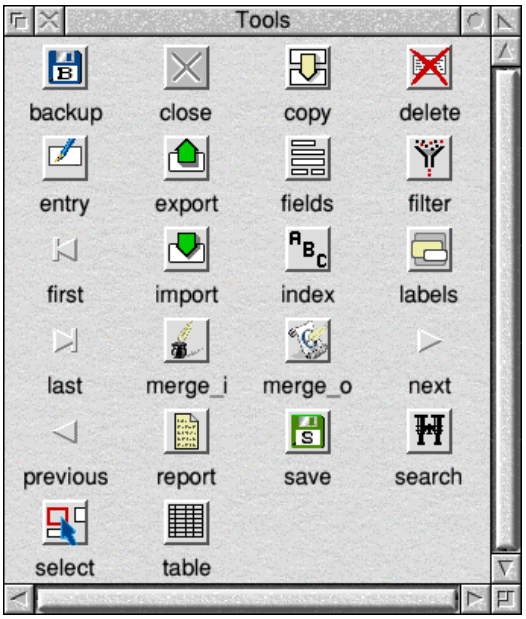
If you do not wish to have a toolbar on your card, you can click with ***Menu*** over the card and choose *Remove toolbar*. To add a toolbar, you can either drag the *Toolbar* icon from the **Field selection window** or choose *Add toolbar* from the card menu. A default set of tools is supplied when a toolbar is placed on the card.

Clicking with ***Menu*** over a tool will open the **edit tool menu** shown below. *Card* leads to the **card attributes window** described earlier in this chapter.



To remove any tools you do not want, simply drag them off the toolbar, or choose *Remove tool* from the *Tool* sub-menu.

To add new tools, choose *Tools...* to open the **Tools** window, or just double-click anywhere over the toolbar area. Drag the tools you require to the toolbar.



You can change the order of tools on the toolbar simply by dragging and dropping them in the desired position. You can insert or remove spaces to the right of a tool by choosing *Insert space* or *Remove space* from the *Tool* sub-menu.

Each tool has help text which may be edited by choosing *Edit help...* from the *Tool* sub-menu. Note that unlike for data fields themselves, the toolbar help applies to all databases. As the file is currently stored within the *!Impact* application directory, customisation of the help text may be lost when upgrading to a newer version of the software.

The *Edit script* option is for advanced use only. It allows you to redefine what the tool does, but affects the action of the tool across all databases.

Chapter 13: Field types

This chapter gives a description of all available field types and their general use, together with descriptions of other items which may be placed on a card.

How to set the various attributes of individual field types is described in *Chapter 14: Changing field attributes*, while a summary of the disc space required, and the range of data that can be stored, is provided in *Appendix 1: Field types*.

13.1 Why have field types?

You may think at first that you only need to store text, and you could indeed build a database using only this field type. However, you could not, for example, sort records into date order if they were entered as text (August would come before January, and the 31st would come before the 2nd).

However, there are other good reasons for the many field types available in Impact. For example, **Menu** fields allow a user to quickly select an item from a pop-up menu, and be confident that there are no spelling errors which would foil a search operation.

Using the full spectrum of field types also allows you to create data cards which are fully consistent in style and integrate well with other desktop applications.

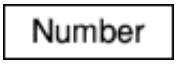
The available types, and their general use, is summarized below.

13.2 Date/time fields

Date/time fields allow a date or time value to be stored and displayed in almost any format, from a simple 23.11.95 to 23rd November 1995. You can specify a different format for each date or time field.

Using date fields instead of text fields allows records to be sorted in date order, and allows **filters** to be used to select records having a specified range of dates. It also allows **calculations** to be carried out on these fields.

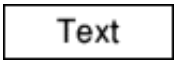
13.3 Number fields



A number field may contain any numerical quantity, and is used for things like money amounts, a person's age, a club membership number, etc. The format in which the number is displayed may be selected to show a specific number of decimal places if required, or none at all.

It can show zero quantities as blank or zero, and can show negative amounts in red.

13.4 Text fields



This is probably the most common data type, consisting of a single or up to 32 lines of text, and is often used to hold names, addresses and telephone numbers, for example. The text entered is stored in a fixed size block on the disc, therefore the amount of space needed must be specified. A trade-off must be made between wasting disc space and having insufficient space for any individual entry.

13.5 File fields

File fields allow you to link to files anywhere on your computer or to store data of any length and format in a separate file inside the database. The field appears on the card as an action button.

There are two types of file field:

- **internal file** (formerly called "Notes"). The file is stored inside the database structure.
- **external linked file**. In this case the field stores the full pathname of the file, which may be on any disc accessible from your computer.

For internal files, you may create them either by dragging a file and dropping it on the button — this will create a copy of the original file — or by clicking on the button to create a new file. A new file is made as a copy of a **template** or sample file, which you can supply. Without a template file, Impact will provide you with an empty plain text file to edit. One advantage of using a plain text file is that it may be merged into an *Impression*/*Ovation Pro* document, like any other field on the card, and the content may be shown in tables. Full details of how internal file fields are handled in record editing are given in *7.6 File fields* while setting up a template for the field is explained in *14.14 File and image fields*.

For external linked files, the link is created by dragging any available file, directory or application from the desktop and dropping it onto the button. The path to the file will be recorded in the database record. Note that if the file is subsequently moved or deleted, Impact will not be able to find it.

A file field displays its legend in black when there is a link to a file, in grey when there is not, and in red when the file cannot be found. When a link has just been created as described above, the legend will be in green until the modified data is saved.

Clicking the button has the same effect as double-clicking on the file in its Filer window. This action will depend on the particular file type. For instance, a text file may be loaded into *!Edit*, a photograph into an image viewer or a sound file played. The legend on the button may be chosen to reflect its intended use.

If you click on a file button whose legend is red, you will get an error message telling you what file it was linked to. To avoid irritating repetition of errors when a disc cannot be found, Impact keeps a note of which discs cannot be found and does not attempt to access the same disc again. If the disc becomes available later but the file button is still coloured red, you need to click on the button to get Impact to check for that disc again.

For more information about use of file fields over a network, see *24.3 Using Impact over a network*.

If you have a file or field in a **table display**, the table will show

- the filename of an external linked file;
- the content of an internal file if it is a plain text file;
- the filetype of any other type of internal file.

13.6 Image fields

Image fields are exactly the same as file fields, except that the image is displayed on the card instead of a button. It will be scaled to fit the display area, preserving the aspect ratio. If Impact cannot render the image itself, it will display the filetype icon for the image instead. In either case you may double-click on the image to load it into another application. Images may also be exported by drag and drop like other fields.

For images which are external linked files, if the file has been deleted or moved a warning triangle will be displayed on the record card. If you double-click on the warning triangle, you will get an error message telling you what file it was linked to. If the file is located on a disc which is not currently accessible, a question mark in a red circle is displayed.

You may change between image and button view for a file field by editing the card.

13.7 Flag fields

A flag field is displayed as a text legend with an associated tick box, and may be used to indicate any on/off option, such as whether a club member has paid this year's subscription. The legend shown may be edited to suit.

A single flag field may consist of up to 32 individual buttons, each with their own legend, and each may be selected independently of the others. This is in contrast to the **option** field type, described below, where only one button may be selected at a time.

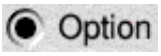
Flag fields may be used within calculations where they take the value of 1 (ticked) or 0 (unticked), and can have many uses in such a context. This is explained in more detail in *16.15 Using flag fields*.

They are also useful within filters, so that records having particular flags set, or not, may be selected. See *4.3 Calculation filters* for details of how to create filters on flags.

When exported, the legends from the ticked flag fields are treated as text, with commas between.

13.8 Option fields

These are displayed as a series of up to 32 radio buttons, and are used to select one item from a range of mutually exclusive options. The displayed legends may be chosen to suit their use.



When merged, or exported, these fields are treated as text, the legend from the selected button being used.

Instructions on how to change the order of option legends can be found in *24.6 Changing the order of texts in option, menu and browser fields*.

13.9 Browser fields

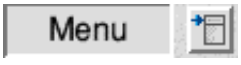
This field type consists of a display icon and a pair of arrow icons. Clicking the arrows allows one of any number of user defined texts to be selected. This could be used to select, for example, from a list of personnel titles such as "Mr, Mrs, Ms", etc., or from a list of club membership categories.

When merged, or exported, these fields are treated as text, the currently displayed choice being used. They are particularly useful where there are a large number of options which would take up too much space on the card if implemented as Option buttons.

Note that it is not possible to insert additional choices part way through the list of texts: any additions must be made at the end of the list. You can choose to have the texts displayed either in alphabetical order, or in the order they were created. If you need to change the order of texts in another way, please see *24.6 Changing the order of texts in option, menu and browser fields*.

13.10 Menu fields

A menu field is identical in use to the browser type described above, except that the available choices are displayed in a pop-up menu invoked by clicking the menu icon.



This field type has the advantage that all the choices are visible at the time of selection, but may not be useful where more than twenty or so items are to be made available, as the menu size may be too long. You can change a menu field to a browser field or back very easily if you

decide later on that the other type will suit you better.

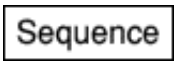
13.11 Calculation fields

The calculation field type may be used to display the results of a calculation made using other fields from the card as required. The result may be a date, number, or text. The calculation is done whenever the card record is displayed, or when the field is included in a report merge operation or exported, etc.

Uses include computing and displaying VAT on a sale, the age of a person based on their entered date of birth, etc. Calculations are fully described in *Chapter 16: Calculations*.

13.12 Sequence fields

A sequence field will display a number which is automatically incremented for each new record created. The next value to be used may be set at any time, and an optional prefix and/or suffix may be specified.



It is a text field with a fixed width of 12 characters. It is set whenever a new record is created, but may be edited at any time later. It is most useful for assigning unique codes such as membership numbers or serial numbers. But note that Impact does not prevent you from editing two records later so that they have the same value in a sequence field.

13.13 Action buttons

An action button is not really a database field, as it does not store data, When clicked, a pre-defined sequence of commands may be executed, to carry out complex operations at the click of a button.

For example, indexes, filters and field maps may be selected, labels or reports printed, field values changed, etc. The commands available are fully described in *Chapter 17: Action commands*.

An action button can also be represented by a menu icon, where clicking on it will open a menu and the action resulting depends on the user's menu choice. You can see an example of this in the *Videos* sample database, where choosing from the Category menu places a code in the box and alters the hire fee accordingly. A menu-style action button could also be used to make it easier to choose foreign data in a relational database: an example script is given in *Chapter 15: Relational databases*.

Action scripts may be edited either from the card editing window or by clicking with ***Adjust*** on the action button when viewing any record. Note that the button must have been given a name before a script can be created.

13.14 Labels

These are not database fields, but when placed onto a card may be used to label real fields to show their use. They may also be used as card titles, or in fact be placed anywhere for any reason.

13.15 Boxes

These are not database fields, but may be used to group related fields together, or for any other decorative effect.

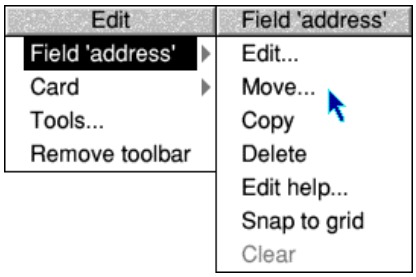
Chapter 14: Changing field attributes

All fields may be adjusted to appear and perform in a variety of ways. For instance, a number or date may appear in a specified format, a text field may have a defined number of lines, and contain a specified maximum number of characters.

This chapter describes how to set these various attributes.

14.1 The Field editing menu

If you click with ***Menu*** over a field or a selection of fields and follow the first item on the **edit card menu**, the **field editing menu** is opened.



If over a single field, *Edit...* will open the **field attribute box** for that field, enabling you to set attributes as described below.

If over a selection of fields, *Clear* will deselect all of the fields.

Move... opens the **field move tool** described below.

Copy, *Delete* and *Snap to grid* do what you would expect.

Edit help... will open a window for you to type a description of the field for use in pop-up help.

14.2 Common features

All field attributes are set when Impact is in **edit mode**, either while designing a new database, or when editing a card layout. These actions are fully described in earlier chapters.

Once placed, fields may be selected, singly or in groups, by clicking with the ***Select*** and ***Adjust*** buttons in the conventional manner. Selected fields will be displayed with a red border, and a red outline box will be shown around multiple groupings. Several actions may be carried out on a selected group of fields. (Note that text fields cannot be extended to create multi-line fields in this manner. These fields are fully described later.)

A selected group of fields may be moved together by dragging with ***Select***, or be extended, horizontally or vertically as appropriate, by dragging with the ***Adjust*** button. For fine adjustment, you can move selected fields around the card by using the cursor keys. Holding ***Shift*** and pressing the cursor keys will move the objects according to the configured grid spacing.

In addition, a pop-up menu may be displayed by clicking on the group with the ***Menu*** button, and this allows the field or group to be copied, moved, or deleted.

14.3 Editing legends

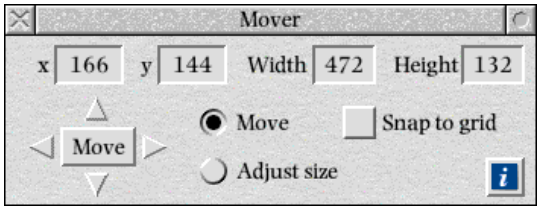
Several field types, such as **file**, **action**, **flag** and **options**, have a text legend which may be edited directly on the card by clicking once with ***Select***. When you do this, the legend will be given a white background and the caret will be placed within it ready for editing in the normal manner.

Pressing the ***Return*** key will return the field to its normal appearance, as will clicking on

another field.

14.4 The field move tool

If you choose *Move...* from the *Field* sub-menu, this will open the **move/adjust** tool shown below.



This tool allows fine adjustment of the field, or selected group, to be made, by clicking on the arrow icons to move the field in the appropriate direction. Clicking with ***Adjust***, of course, reverses the direction.

For fast moves, holding down either ***Shift*** key will cause the field to move in larger increments, and all the arrows will auto-repeat if the mouse button is held down.

The radio buttons may be selected to toggle the tool action between **move** and **adjust**, the button legend changing to reflect the current mode. In adjust mode, the appropriate side of the field will be extended when an arrow is clicked. The ***Adjust*** mouse button will reverse the action and reduce the field size on the appropriate side.

Note that some field types may only be extended horizontally and not vertically, and clicking with the up or down arrows in this case will do nothing (except to cause the computer to beep).

When adjusting a group of fields in this manner, some may adjust in both directions while others will not, and this should be taken into account when making a selection of fields for this purpose.

Snap to grid if ticked will cause the field(s) to be moved or resized in steps which fit Impact's set grid pattern.

14.5 The field edit box

Double-clicking on a field or clicking with ***Menu*** over the field and choosing *Edit...* from the *Field* sub-menu will open an **field attribute box** for that field. Many of these have similar features detailed below.

14.6 Field names

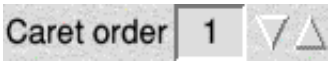
Most field attribute boxes have a writable icon to allow a field to be given a name. Field names are used by Impact when offering a choice of fields in a menu, as column headings in a **table display**, and in **relational links** to other databases.

It is not necessary to name all fields, and a database may be created without doing so. However, un-named fields will not be shown in tables, or on field menus, and therefore cannot be exported, merged, or used in calculations etc. Unnamed fields may not be searched using the search tool.

Any unnamed field may be given a name later as required. Field names cannot be changed after being defined. Once a name has been entered in the writable icon provided, it will become permanent, either when you press the ***Return*** key, or when the attribute window is closed by any means.

It is recommended that you use lower case letters for your field names, as this will avoid clashes with current or future names of functions, which are used in calculation expressions.

14.7 Order or caret order



Most fields may also be given a **caret order** (sometimes just referred to as **order**). This is a number indicating the order in which the caret will visit writable fields during data entry and editing.

The caret order may also be used to determine the order in which fields are shown on field menus, but these may be displayed in alphabetical order, according to the settings in the **Choices** window (see *Chapter 21: Choices*).

Caret order numbers are allocated on an increasing basis as fields are placed, but may be changed at any time. They do not have to be sequential, and two fields may have the same value. In this case the actual ordering will be arbitrary.

14.8 Default entries

Most fields may be given a default entry, that is, data which will be displayed automatically when a new record is created.

In writable fields such as **text**, **dates** and **numbers**, the default entry is entered as an expression which will be evaluated and placed in the field. This means, for example, that default text should be entered between double quotes. Calculations and expressions are fully explained later.

If the box is not long enough for the expression, you can ***Ctrl-***double-click to load the expression into your text editor via OLE (Object Linking and Embedding). After making changes, saving the file will transfer it back into Impact.

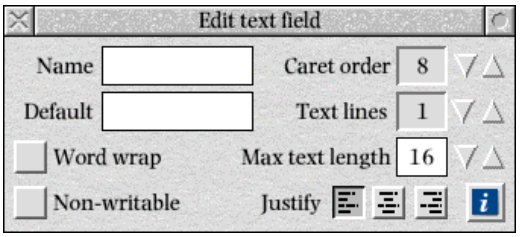
14.9 Making writable fields non-writable

Writable fields such as **text**, **dates** and **numbers** can be set to non-writable by ticking the *Non-writable* flag. This is useful where you wish the field to be updated by an **action script** rather than by a user typing.

14.10 Justification

For many field types the text or value in that field may be left-, centre- or right-justified within the field by selecting the appropriate justify icon.

14.11 Text fields



Max text length

An important attribute of a text field is the maximum number of characters it is required to hold, and this should be set as required using the adjuster arrow icons, or by entering the value directly. The maximum length can be anything up to 999 characters.

A trade off must be made between allocating sufficient space, and wasting disc space. The space occupied on disc for a text field is the same for each record in the database. It is better to keep the space allocation to a minimum, as this may always be increased at a later date, if more

space is required.

Note that the maximum number of characters set and the size of the text field icon are unrelated quantities. In the case of multi-line text fields, there may be far more space available in the display than will ever be used in practice.

Multi-line text

The size of a text field's box may be increased up to a maximum of 32 lines by adjusting the *Text lines* setting, although between 1 and 6 is more usual. If you require more, consider using an **internal file** field instead. The available space (as discussed above under **Max text length**) will be distributed over the lines as required when text is entered or displayed. In some circumstances all the space might be used in a single line.

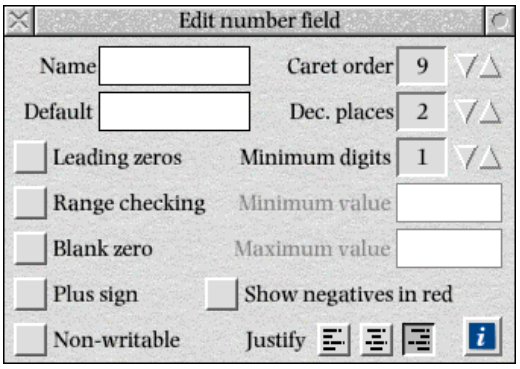
Multi-line text fields may contain more lines of text than set in *Text lines*, in which case a scrollbar will be added to the field so that you can view the whole content. However if *Text lines* is set to 1 then the field is restricted to only containing one line of text, and if multi-line data is imported the new line characters will be converted to spaces.

Note that if you have a multi-line text field in your record card, then when the record is displayed on a single line in a table or in a report the lines of the multi-line field will be concatenated with commas. When using the export tool you can choose whether to separate or keep multi-line fields together.

Word wrap

Selecting the *Word wrap* option on a multi-line text field will cause text to be wrapped onto subsequent lines during entry.

14.12 Number fields



Decimal places

The required number of decimal places may be set using the adjusters provided by *Dec. places*. If only whole numbers are required this may be set to zero, in which case no decimal point will be displayed.

Minimum digits

These adjusters set the minimum width of the displayed number, and is most useful in conjunction with the **leading zeros** option described below. It specifies the minimum number of characters to be used for the whole value, including the decimal point and decimal places, as well as any sign required or specified.

If the value requires fewer than the specified number of digits, it will be padded with leading spaces up to the given number. If the value requires more characters, these will be used.

Leading zeros

If this option is selected, and the minimum number of characters has been defined as above, then any unused characters will be filled with leading zeros. Using this feature, numbers may be displayed in a fixed width, such as '00074' for example.

Range checking

With this option selected, an entered value will be checked when an attempt is made to save the record to disc, and if outside the bounds of the range given, an error message will be generated.

Selecting this option will cause the *minimum* and *maximum value* boxes to become writable, allowing the limits to be entered.

Blank zero

If this option is selected, a zero value will be displayed as a blank field, rather than as zero digits.

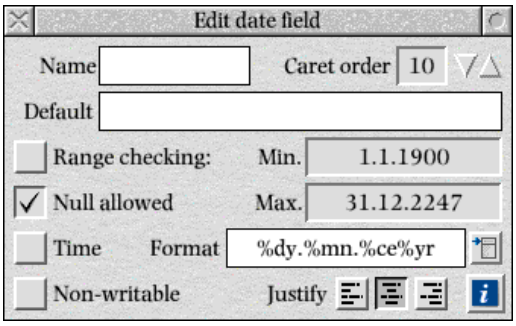
Plus sign

If this option is selected, the displayed value will always show a sign; plus or minus as appropriate. A negative sign is, of course, always shown when required.

Show negatives in red

If this option is selected, all negative numbers will be displayed in red, rather than black. This only applies to the card display, and does not affect merged or exported values.

14.13 Date/time fields



Range checking

Selecting this option will cause the *Min.* and *Max.* fields to become writable, and allow the entry of an acceptable date range. With this option selected, a check will be made when an attempt is made to save the record, and an error generated if the entered value is outside the range specified. With this option not selected, dates will be checked for their value being between 1st January 1900 and 31st December 2247, as only this range of dates can be stored in a date field.

Null allowed

If this option is ticked then it is permitted to leave a date field blank.

Date or time format

The required date format may be entered directly into this field, or the menu button may be

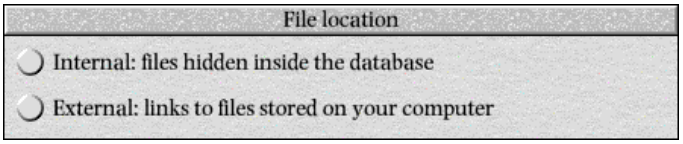
clicked and a suitable format chosen from the menu offered.

Date fields may be tagged as time fields by ticking the *Time* option. If this is done an alternative menu of formats will be offered. Times are stored in the same way as full dates, and the formats may be mixed. However, setting this option ensures that entered data is interpreted as a time and not a date.

The pre-defined formats offered on the menu are taken from a file within the Impact application directory, and may be edited by the user to provide alternative selections. The format definitions are described in *Appendix 3: Date format codes*.

14.14 File and image fields

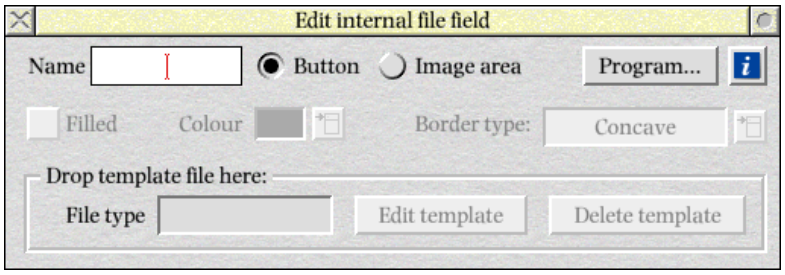
When a file or image field is first placed on a card, you must choose whether the files are to be kept inside the database, or linked to elsewhere on your computer, using the dialogue box shown here.



Once the field has been created it cannot have its storage arrangements changed. However, you may switch between button and image display by editing the field attributes and clicking the appropriate radio button.

Image fields may be given a border by selecting a style from the *Border type* menu. The available styles include a plain line and standard 3D styles. If the Filled option is selected, the field background will be filled in before the image is drawn, using the colour you have selected from the menu provided.

Internal file fields - Template files

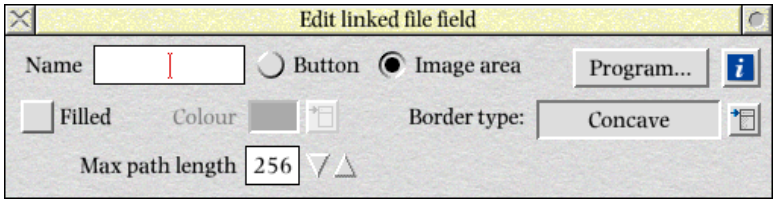


As described earlier (see *Section 13.5*), if you click an internal file field button and there is no file stored in the field, Impact creates a new file by making a copy of a stored template file. If no such file is provided, Impact will create an empty plain text file.

To specify an alternative file type, say an *Ovation Pro* or other DTP file, simply create a file (it may contain text if required), and drop it into the *File type* field on the above dialogue box. The file will be stored within the database, and the file type will be indicated in the field. This file will then act as the template.

The *Edit template* button will be enabled if you have provided a template file, and will open the template file in the appropriate application to allow you to make alterations. Templates may be removed using the *Delete template* button.

External file fields - Max path length



A linked file field holds the full pathname of the file, directory or application dropped onto it, and sufficient space on the disc must be allocated to hold this. If your file is held in a deeply nested directory structure you may need to increase this allocation.

This can be increased at a later date, if problems arise, such as the Filer complaining it cannot find the file, or that the file name is invalid. These errors may be due to the file name having become truncated.

Program

It is possible to invoke an **action script** when a new file is dropped on an internal or external file field. Note that you must give the field a name before you can create the action script.

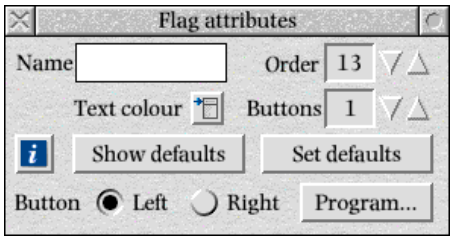
Clicking on the *Program...* button will cause *!Edit* (or your configured text editor) to display a text file window. If you are editing an existing card field, this window will show the current commands.

Action scripts are fully described in *Chapter 17: Action commands*.

Having entered or edited the commands, the text should be saved to its existing filepath within the database. If you have only changed the action program on the card, there is no need to also use the *Save* button to store the new action.

You might use such an action to verify the content of the file or to do other processing, such as setting up field values based on the content of the file.

14.15 Flag fields



Buttons

This adjuster allows the number of flag buttons to be selected. These are created as you adjust the value, immediately below the placed field. If you require flag buttons placed elsewhere on the card, you must create a separate field.

Default selection

To choose which flags are selected by default, tick the boxes on the record card which you require, and click on *Set defaults*. Click on *Show defaults* to show on the record card which ones are selected by default.

Button position

Which side of the text the button appears may be selected by clicking the *Left* or *Right* option

button.

Text colour

This may be chosen by clicking on the *Text colour* menu icon and selecting the colour required. It is strongly suggested that black is used, unless there is some over-riding need, perhaps where some colour coding is important on the card.

The colour will be applied to the text associated with the button which was originally clicked on to invoke the dialogue box.

Program

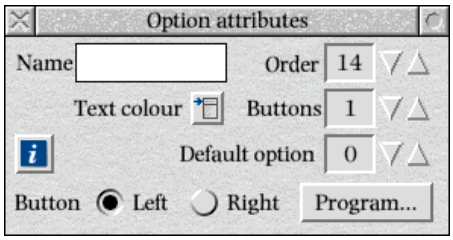
It is possible to invoke an **action script** when choosing a flag field. Note that you must give the field a name before you can create the action script.

Clicking on the *Program...* button will cause *!Edit* (or your configured text editor) to display a text file window. If you are editing an existing card field, this window will show the current commands.

Action scripts are fully described in *Chapter 17: Action commands*.

Having entered or edited the commands, the text should be saved to its existing filepath within the database. If you have only changed the action program on the card, there is no need to also use the *Save* button to store the new action.

14.16 Option fields

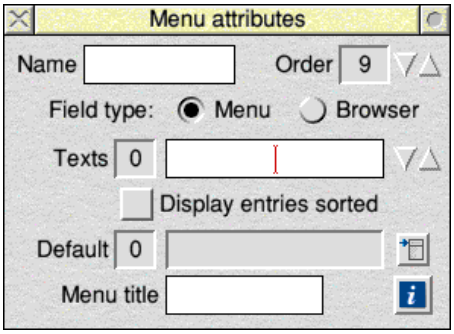


Option field settings are identical to those for **flag fields** as described above, except for the way default settings are chosen. Instructions on how to reorder texts can be found in *24.6 Changing the order of texts in option, menu and browser fields*.

Default option

Choose from the *Default option* menu which option should be selected by default. The options are numbered sequentially downwards beginning with 1. A setting of 0 will result in no button being pre-selected.

14.17 Menu and browser fields



Choosing the field type

Menu and browser fields are very similar types: the only difference is in how the values are entered into the record cards, via a menu or via arrow nudge buttons. Unlike other field types, you can easily switch a field from a menu type to a browser field and back, by using the radio buttons.

Entering texts

Texts used in the menu are numbered from zero upwards, and should be entered into the writable icon. Pressing the ***Return*** key will cause the number to be incremented, allowing the further items to be entered.

If you require a blank entry to be available in the database, note that you will need to create an entry with some text in, create a further entry, and then go back and remove the unwanted text from the one you wish to be blank. The blank entry must not be the last one in the list, as this will be ignored when the menu / browser list is constructed.

The up and down arrows may be used to select any entered text, which may then be edited if required. You will always find one extra blank text at the end of the list, to allow additional texts to be entered.

The texts will be displayed in the database in the order in which they were created, or you can tick *Display entries sorted* to have them sorted into alphabetical order. Note that additional texts may only be added to the end of the list, and apart from choosing alphabetical order, the order of texts on the menu may not be rearranged easily. Instructions on how to reorder texts can be found in *24.6 Changing the order of texts in option, menu and browser fields*.

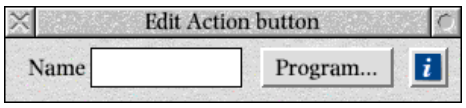
Default entries

The *Default* menu may be used to select the entry to be displayed by default, when entering a new record. If you require a blank field to be selected, this should be provided in the text list entered.

Menu title

The menu may be given a title by entering this in the writable icon provided.

14.18 Action buttons



Programming the action

Clicking on the *Program...* button will cause *!Edit* (or your configured text editor) to display a text file window. Note that you must have given the action button a name before you are able to do this. If you are editing an existing action button, this window will show the current script commands.

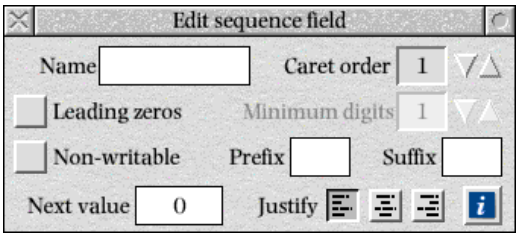
Action script commands are fully described in *Chapter 17: Action commands*.

Having entered or edited the commands, the text should be saved to its existing filepath within the database. If you have only changed the action program on the card, there is no need to also use the *Save* button to store the new action.

You can also edit an action script by clicking with ***Adjust*** on the action button when viewing a record card.

14.19 Sequence fields

Sequence fields are limited to a total of 12 characters. This must be borne in mind if you choose to add a prefix or suffix to the field, as if the value to be inserted in a new record is too long, it will have to be truncated.



Leading zeros

Selecting this option will cause the *Minimum digits* selector to be un-faded. This is described below.

Minimum digits

If *Leading zeros* is selected, this controls the minimum number of digits used to represent the sequence number, excluding any prefix or suffix. Unused spaces will be filled with zeros.

Next value

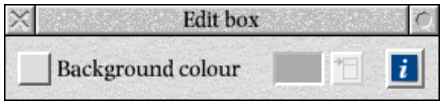
This field indicates the next sequence value to be used, and may be edited at any time. If you are creating a new database, enter the first value you require here.

Prefix & Suffix

These fields optionally contain a text which will be used to precede or follow the number when entered into the field. If you want a space to appear between the prefix and the number, you must add this explicitly at the end of the prefix, and similarly for spaces preceding the suffix.

14.20 Box items

These are not real data fields, but may be used to group fields together in a logical manner.

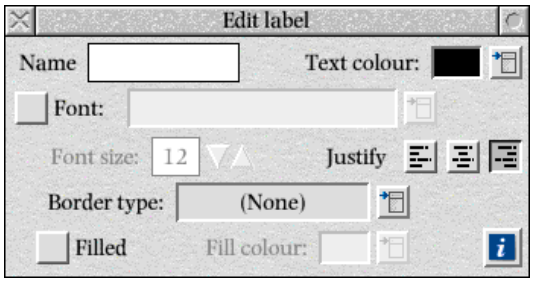


The Box attribute dialogue box permits the choice of a background colour if required. Colour should be used with discretion.

14.21 Label items

Like boxes, these are not real data fields, but may be used to provide descriptive legends anywhere on the card. They are most useful for labelling writable field icons, or for providing a title for the card.

They may have a border, and be filled with colour. However, in normal use they should appear black without a border. Colour should always be used with discretion. Remember that some people are colour blind, while others may be using a monochrome display where colours may not be easily readable. Use colour where it conveys some purposeful meaning.



Name

It may at first seem strange that a label can have a name like a real data field. However, it is possible to assign a text to a label within an **action script**, to change the label text under certain circumstances.

We suggest you leave the label un-named until the need arises.

Text colour

Text colour may be selected from the menu.

Font

If you wish to use anything other than the standard desktop font for your labels, tick the *Font* box and choose a font from the menu, and the font size you require.

Justification

Normally a label for a writable field will be placed on the left of the field, and be right justified. A title panel may require centre justification. Select the icon required as shown above.

Border type

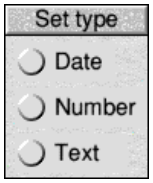
You can select a border style from the *Border type* menu. This may be useful for creating a title panel for a data card.

Fill colour

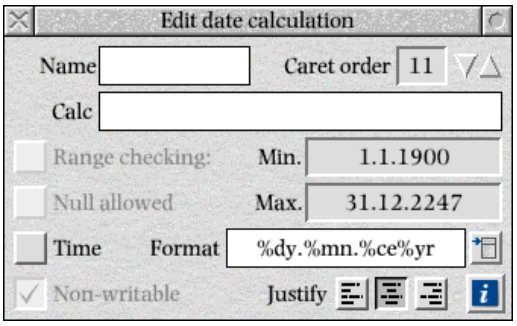
Select the *Filled* option if you require the label to have a filled background, the colour of which may be selected from the *Fill colour* menu provided.

14.22 Calculation fields

When a calculation field is first placed on a card, you must select the calculation result type you require by selecting from the options in the dialogue box shown here.



When you now double click on the calculation field, the attribute dialogue box for the field type selected will be displayed, allowing type-dependent attributes to be set. Refer to the **date**, **number**, or **text** field types above for details of how to set these types.



In each case, the required calculation should be entered into the *Calc* field on the appropriate dialogue box. Remember that the fields on a card, including calculations, are updated in **caret order**, and this may affect the calculation result. You would normally place calculation fields last in order.

If the box is not long enough for your calculation expression, you can ***Ctrl-***double-click to load the expression into your text editor via OLE (Object Linking and Embedding). After making changes, saving the file will transfer it back into Impact.

Calculations are fully explained in Chapter 16: *Calculations*.

Chapter 15: Relational databases

This chapter describes what relational databases are, and how they may be implemented within Impact.

15.1 An explanation

A **relational database** is the term used when data is stored in more than one database, and where the data in one database is related to the data in another.

For example, a database containing a list of videos, used in a video library, might include a field showing the customer currently renting the video. Since one customer may be allowed to rent several videos at one time, it would be very wasteful of disc space if the full customer details were entered for a every video rented.

Also, if a customer's address required updating, it would be necessary to find all references to that customer, and ensure that they were all updated correctly. This is clearly not efficient.

Instead, a separate customer database is kept, so that each video database record need only have some reference to the relevant record in the customer database. In this way, the customer database could be referred to by many other databases, while keeping the customer details all in one place.

15.2 Foreign fields

In a relational database system, Impact uses the following terminology. Where the video data card may contain a copy of the customer's name field from the customer database, then the video database is termed the **local database**, which contains a **foreign field** (the customer's name). Similarly, the customer database is referred to as the **foreign database**.

It is thus possible for the local (video) database to contain many references to the foreign (customer) database. Impact places no limit on the number of databases which may be related in this way, and a local database may reference a foreign field which is itself a reference to a further foreign field.

Key fields

Even if it is required to include several fields from the foreign database in the local data card, such as the address fields in addition to the customer's name, only one link field is needed, and this is referred to as the **key field**. This is the field which contains the link to the foreign database record.

Dependent fields

Other fields, such as the address fields, are known as **dependent fields**, as they rely on the key field to provide a reference into the foreign database.

The difference between a key field and a dependent field is that the key field establishes the link between the two databases. The dependent fields are simple copies of the foreign database fields, displayed in the local database card as required.

15.3 Creating foreign fields

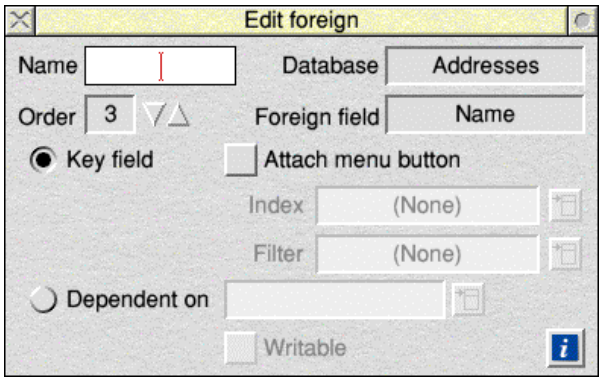
Impact provides a simple, and logical way of including foreign fields in a card. All that is necessary is to drag the required field from the foreign database (the customer database in the above example), from its card to the card being designed.

When first placed, all foreign fields are assumed to be key fields. Dependent fields are formed by giving them a reference to a key field from the foreign field attribute dialogue box, as

explained below.

15.4 Foreign field attributes

Double clicking on a foreign field will cause the following attribute dialogue box to be displayed.



In the above illustration, the *Name* field from the *Addresses* database has been dragged to the local database card being created. The attribute box shows the name of the foreign database and field. The *Key field* radio button is selected to show that this is a key field.

Note that all foreign fields must be given a name, whether they are key fields or dependents. This may be done, as described in *Chapter 14: Changing field attributes*, by entering a name in the dialogue box for the foreign field, as shown above, and pressing the ***Return*** key. Field names must be unique within each database, but there is no reason why the same name cannot be used in different databases.

To include the address fields as dependent fields, drag these onto the card in a similar manner, then click on the *Key field* flag to un-tick it. This will allow you to choose a key field from the menu for this (now) dependent field.

There is no limit to the number of foreign key or dependent fields allowed on a card, and it is quite possible to have (say) two customer's names (key fields) and addresses (dependent fields) on a card, each address being dependent on its own key name field.

Key fields are non-writable, but a dependent field can have the *Writable* flag ticked in this dialogue box, to allow remote editing of the data as described in a later section.

15.5 Type-specific attributes

The local attributes for a foreign field will be the same as the attributes of the field on its "home" card, with the exception of **caret order**, which can be set for its use on the current card.

15.6 Entering foreign data

We now take a look at how to enter foreign data during normal use of the database.

The direct method

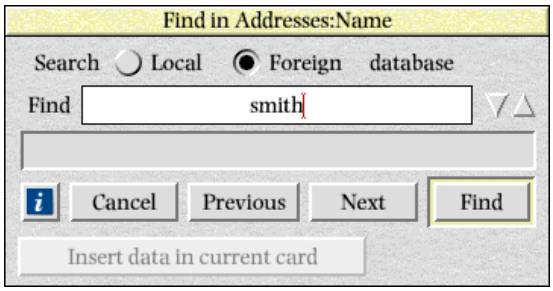
Taking the above example, if a customer hires a video, then the customer's record must be found in the customer database, usually by using the **Find tool** accessed by ***Alt***-double-clicking on the name field.

The name field may then be dragged from the *Addresses* database card onto the key field (the customer's name) on the video database card. When this is done, the name will be copied across and the link established. Dependent fields in the videos data card, the customer's address for example, will also be automatically displayed.

This action will flag the card as modified, and you should save the new data by clicking on the database's **save tool** or pressing ***F3***.

The indirect method

This method does not require the foreign *Addresses* database to be displayed on the screen. If you ***Alt***-double-click over the foreign *Name* field on the *Video* database card to display the Find window, you will find it has extra buttons, as shown below.



When first displayed, the *Foreign database* option is selected. When you click on the *Find* or *Next* buttons, the foreign *Addresses* database will be searched, and the first Smith found will be displayed in the grey box. If this is the customer you wish to use, click on *Insert data in current card*. If not, use the *Next* and *Previous* buttons to browse the *Addresses* database for further Smiths.

This action is directly equivalent to dragging the chosen customer from the *Addresses* database card, without the need to have this card displayed, and is generally the preferred method.

Selection from a menu

With Impact version 3.39 onwards it is possible to attach a menu button to foreign key fields. This is done by ticking the *Attach menu button* option in the field attribute dialogue box. Clicking on the menu button will open a menu showing the foreign records. The user can select the record to link to very easily by this method.

The field attribute dialogue box also allows you to select an **index** and a **filter** which will be used in constructing the menu. These are each taken from the indexes and filters defined in the foreign database. The index will determine the order in which the records are displayed on the menu. Choosing a filter allows you to limit the records from which the user can select to a subset of those in the foreign database. (An index might also restrict the records shown in the menu, if it is based on a selection or includes a filter in its definition.)

Note that you can always access the full range of foreign records by using the direct and indirect methods of editing described above. The index and filter will only limit the records shown in the menu.

Using an action script to generate a menu

Prior to Impact version 3.39 if you wanted to choose the foreign data from a menu you would have had to use an action script. In most cases it will now be simpler to attach a menu by using the option in the foreign field attributes. However, in some cases there may be particular need to use an action script for greater control, so this example is still retained.

Suppose, for example, you had a database of railway photographs and a related database containing the names of the railways. Using a menu-style action button you could pick the railway in the photograph without having to search for it in the related database. Here is an example of the script you might use. Note the use of the **AddSlashes** function to escape any commas in the railway names.

```
Local "list", "choice";         !Set up local variables
Index Railways:"sorted";        !Turn on index to sort railways
First Railways:;                !Start from the beginning
list=AddSlashes(Railways:name,",");         !First menu item
While (Next("Railways")) {      !Move through Railways d-base
  list+=","+AddSlashes(Railways:name,",");  !Add to the list
}
choice=Menu("Railway",list);    !Put up menu and get user's
                                !choice
if (choice=="") Abort;          !Abort if user chooses nothing

First Railways:;                !Go back to Railways d-base
Find Railways:"name",choice;    !Find the record chosen
railway=Record("Railways");     !Set value in our foreign key
                                !field
```

Note that this method is only reliable if the values in the key fields are unique, so would not be so much use for linking to a database of people, where two might easily have the same name. To avoid this problem you might need to include the foreign record number in the value displayed on the menu, and extract that from the user's choice.

15.7 Remote editing

Foreign dependent fields may be edited remotely if they have been flagged as writable in the **foreign fields attributes box**. This permits the customer's address, for example, to be updated from the video record card if required, without the need to display the customer database.

When the local record in the *Videos* database is saved, the appropriate fields in the related *Addresses* database will be automatically updated at the same time as the local database fields.

The key field (the customer's name) cannot be edited remotely.

15.8 Clearing foreign fields

Since data is entered into these fields by dragging, and the key field is not editable, these fields cannot be cleared in the normal manner. Instead, click with ***Menu*** over the key field and follow the arrow by *Clear item* to display a window showing the current field contents. To clear the field, simply click on the *Clear* button. This will result in the link to the foreign database record being broken. Any dependent fields are also cleared.

This renders the card modified, and the **save tool** must be used to permanently save the change to disc.

15.9 Opening the record in the remote database

Although you may have several fields from the remote database displayed in the local database's record card, it is often still necessary to view or edit the record in the remote database using the full record card for that database. For example, the *Videos* database displays the name of the borrower as the key field, and the address of the borrower as a dependent, but to obtain the postcode or phone number of the borrower you will need to go to the *Addresses* database itself.

Impact provides a convenient way to open the right record in the remote database. Simply click ***Ctrl-Select*** on the key field, and the remote record card will be opened at the correct record.

Chapter 16: Calculations

This chapter describes **calculations**, which are used in several contexts within Impact, such as default data entries for writable field types, merged calculations, and action script programming.

16.1 Expressions

Mathematical expressions are used in the definition of calculation filters, default field contents and calculation fields, and may be used within action scripts.

An expression is a mathematical formula using operators (+ - etc.) and operands. A full list of operators is given later. Operands may consist of constants, field names, pre-defined functions, and user defined variables. These may represent numeric values (integer or real), dates, times and text strings.

Spaces may generally be used in expressions between operators and operands if desired, but note that in some circumstances, such as specifying a temporary calculation filter in a merge command, spaces are not permitted.

16.2 Syntax definitions

The syntax definitions used below are expressed using the following conventions.

Optional items are shown within square brackets: `[optional-item]`

Items within angle brackets are to be replaced by the actual item required. For example, `<field>` should be replaced by the actual field name to be used, excluding the brackets.

Other items should be included literally.

In the examples given for some functions an arrow ➔ is used to indicate the result of the expression given. For example:

```
Text("abcdefghij",4,3)    ➔ "def"
```

indicates that the result of the expression `Text("abcdefghij",4,3)` is `"def"`

16.3 Constants

Constants are recognised by their format. Text constants must be enclosed within double quotes ("), and numeric constants must start with a digit. Impact cannot recognise date constants as these can be expressed in a variety of formats. However, literal dates can be included by using the **Date** function.

Examples:

Text string constant	`"Hello world"`
Numeric constants	`234.56`
	`2.3456E3` (scientific notation)
	`0x2AF` (hexadecimal number)
Date constant	`Date("15th May 1999")`

16.4 Control codes in text strings

Text strings used in calculations and action commands may include control characters, prefixed by \ (backslash). Allowed codes are:-

`\n`	newline
`\r`	carriage return
`\t`	tab
`\"`	double quote (to embed quotes into a literal string)
`\x##`	any ASCII character, where ## should be hex digits
`\\`	to insert a \ (backslash).

e.g. `"59, High Street,\nLondon\nN10 2GT"` will split the address over separate lines in a multi-line text field.

16.5 Reading values from fields

The full syntax of a field name is:

```
[<database>:][<key>.]<field>
```

where:

`<database>`	is the name of the database containing the field or key
`<key>`	is the key field in the local or named database providing the link to a foreign database.
`<field>`	is the actual field name in the local, named, or related database.

Examples:

```
phone
```

Represents the field *phone* in the local database.

```
name.phone
```

This uses the local key field *name* and then finds the field *phone* for the related record in the foreign database to which the key refers.

```
Videos:name.phone
```

Since the database name is specifically given, this would refer to the same field as that above, but may be used in a third unrelated database. In this case the record currently displayed in the *Videos* database would be used.

```
Addresses:name
```

Since no key field is specified in this case, 'name' refers to a field *name* in the Addresses database. Again, since this specifies the database by name it may be used in any other database, and will refer to the record currently displayed in the Addresses card. The database name is not needed if field *name* is in the local database.

Only databases stored in the same directory as the current database may be accessed by this method. You cannot, for example, access fields in databases stored in *SampleData* from a database stored in the *UserData* directory.

16.6 Types of expression

Number expressions

These are expressions which result in a numeric value. For example:

```
3.5*(4+5)

cost*(1+vatrate/100)
```

where *cost* and *vatrate* are assumed to be the names of fields on the same card.

Text expressions

These are expressions which result in a textual value. For example:

```
"Mr "+forename+" "+surname
```

Here, the names used are the names of text fields on the card, and the expression will result in the content of these fields being concatenated together, with a space between.

Note that in any expression, unquoted words such as *forename* or *cost* are always assumed to be the names of other fields on the card. In order to include literal text, such as *Mr* and the space in the example above, it is necessary to enclose them within double quotes.

Date and time expressions

It is also possible to have an expression resulting in a date or time. For example:

```
date+28
```

This may, for example, compute the due date for payment of an invoice, by adding 28 days to the invoice date.

Adding a number to a time will result in another time, with the number to be added being interpreted as centiseconds.

Mixing data types

Where an expression includes values of different types, Impact will automatically convert types before performing an operation. For example, in the above date calculation, a date field and an explicit number have been used with an **add** operator.

In these circumstances, values are converted according to a priority list as given below. In all operations the value with the lowest priority will be converted to the higher type before the operation is performed.

Data type conversion priorities:-

Type	Priority
Integer	1
Real	2
Time	3
Date	4
Text	5

For example, the expression

```
"Today is "+date
```

is evaluated by converting *date* to the higher priority text type, and the result is text, for example

```
Today is 24th May 1997
```

When a number is converted to a date it is interpreted as a number of days (as illustrated in the previous section), and when converted to a time it is interpreted as a number of centiseconds.

If a date or time is used as input to a function which is expecting an integer or real type parameter, it will be converted to the appropriate type. Times will be converted to the number of centiseconds since midnight, while dates will be converted to the number of days since 1 January 1900. If a date is converted to a real, rather than an integer, it will generally be converted as the number of centiseconds since 1 January 1900. Putting date values into maths functions may therefore yield unexpected results!

In addition to these type conversions, the result of an expression, regardless of the field types it contains, will always finally be converted to the data type of the field it is displayed in.

Null values

Date and time types can have a *null* value if the datum is unknown. This will appear as a blank entry in a record card and exported data.

In calculations and action scripts the null value behaves essentially as an unknown value. Testing whether it is smaller, larger or equal to anything else (including another null value) will always yield FALSE. Adding days or months to a null date will yield a null date. When converted to a number, a null date yields the value 0 (as does 1st January 1900).

In a simple filter, however, you can test for a date being null by simply leaving the Data box blank. Null dates will not be retrieved if your filter tests for a date being greater than or less than another.

16.7 Functions and variables

Please note that, unlike action commands, variables (and field names) are case sensitive, and must be entered exactly as shown below. All functions start with an upper-case letter. To ensure user defined field names and variables do not clash with current or future functions, we recommend that these contain only lower-case.

Syntax

The definitions of functions given below are presented as follows:-

```
type Function(type a,type b ... )
```

where `type` indicates the data type of the value returned by the function and the types of the parameters required, and may be one of:-

`int`	An integer value (1, 2, 3, etc...)
`real`	A real value, i.e. 12.34567
`date`	A date or time code.
`text`	A text string.

Items in [square brackets] are optional. Where no parameters are required, the parentheses () are optional.

Type conversions

It should be noted that all supplied parameters will be converted to the required types shown below, before the function is evaluated. It is not necessary, therefore, that the parameters be supplied as the required types, as long as they may be converted meaningfully.

16.8 Table functions

The following are assigned values during a **Table** or **Scan** merge command, and may be used to merge column totals and other statistics, following the table, or used in an action script following the use of the **Scan** action command.

Total

```
real Total(int n)
```

The total of column *n*.

TotMin

```
real TotMin(int n)
```

The minimum value of column *n*.

TotMax

```
real TotMax(int n)
```

The maximum value of column *n*.

TotMean

```
real TotMean(int n)
```

The mean value of column *n*.

TotSqr

```
real TotSqr(int n)
```

Sum of the squares of column *n*.

TotStd

```
real TotStd(int n)
```

The standard deviation of column *n*.

Count

```
real Count()
```

The total number of records processed.

16.9 Query functions

These functions all cause the display of a message, question or menu, and wait for the user to

respond. They return the result of the user's choice. It is not permitted to use any of these functions in the definition of a calculation field, the default value expression for a field, or in defining a calculation filter. They are intended for use in action scripts.

Query

```
int Query(text q [, text buttons])
```

This causes a standard message box to be opened containing the given text *q* and *Cancel* and *Continue* buttons. The returned value is zero (false) if the user clicks *Cancel*, and 1 (true) if *Continue* is clicked. This is most useful in an action script, for example

```
If (!Query("Are you sure?")) Abort;
```

A variation of the function allows a second parameter to be passed which is a comma-separated list of texts for the buttons to present to the user. There may be more than two buttons: the maximum number permitted depends on the version of the operating system which you are running.

The first button listed will be highlighted as the default and will appear right-most in the message box. Other buttons will appear in order leftwards. If the second parameter is given, the value returned from the function will be 1 for the first button, 2 for the second, and so on.

```
x = Query("Please choose a colour:", "Red,Green,Blue");
```

In the above example, the buttons will appear from left to right in the order Blue, Green and Red, with Red being highlighted as the default. If the user clicks on Blue then *x* will be set equal to 3.

AskText

```
text AskText(text q [, text d])
```

Displays a query box having question *q* and (optionally) supplying the default reply *d*. The user may edit the reply and/or click *Enter*. Clicking *Cancel* terminates execution of the action script.

AskDate

```
date AskDate(text q [, text d [, text f]])
```

As for **AskText**, but returns a date. The optional third parameter *f* specifies the format the date is to be expected in. If this is omitted, *day*/*month*/*year* is assumed. Date formats are explained in *Appendix 3: Date format codes*.

AskNum

```
real AskNum(text q [, text d])
```

As for **AskText**, but returns a number.

Menu

```
text Menu(text t, text c [, int d])
```

Displays a menu and returns the user's choice, where *t* is the menu title, *c* is a list of comma separated options, and *d* (optional) is the default entry item number, which will be ticked. If the user does not make a choice, the default is returned, if supplied. If any of the items in the list contain commas, these will need to be escaped using the **AddSlashes** function.

16.10 Text functions

These functions manipulate text strings, or produce text from other data types.

DataPath

```
text DataPath([text d])
```

Returns the full file path for given database. If *d* is not given, the current database is assumed (i.e. the one from which the action script is executed).

DataName

```
text DataName([text d])
```

Gives the name of the database specified by *d*. If *d* is not given, the current database is assumed (i.e. the one from which the action script is executed or in which the calculation is evaluated).

FieldText

```
text FieldText(text f)
```

Returns the textual version of the value of field *f* in the current record, formatted as it would be displayed on the card. This is particularly useful for **flag** fields which otherwise yield a numeric value. To read the value of a field which is not in the local database, *f* may include the database name in the form `database:field`

e.g. `FieldText("Videos:type")`

If you wish to read the numeric value of a flag, option, menu or browser field, use the **FieldValue** function.

CardText

```
text CardText(text f)
```

The currently displayed content of field *f* on the record card, whether saved to disc or not. To read the current content of a field which is not in the local database, *f* may include the database name in the form `database:field`

e.g. `CardText("Videos:film")`

Len

```
int Len(text a)
```

The length of the given text string *a*, i.e. the number of characters.

UCase

```
text UCase(text t [, int a [, int b]])
```

Converts *b* characters in text string *t* to upper case, starting from character *a*.

If *b* is omitted, all characters from character *a* onwards will be converted to upper case. If *a* and *b* are omitted, the whole string will be converted.

Either or both *a* and *b* may be negative to count from the end of the string.

LCase

```
text LCase(text t, int a, int b)
```

As for **UCase**, but converts to lower case.

SubText

```
int SubText(text a, text b)
```

Returns the character position of text string *b* within text string *a* with the numbering starting from one for the first character position. A value of zero is returned if *b* is not found.

e.g. `SubText("abcdefghij","def")` ➔ `4`

Word

```
text Word(text t, int n)
```

Extracts the *n*th word from text string *t*. If *n* is negative, extracts the *n*th word from the end, counting backwards. The function returns an empty string if there are fewer than *n* words in the string. A word is considered to be any sequence of characters other than the space character.

e.g. `Word("This is a string",-1)` ➔ `"string".`

Text

```
text Text(text t, int s [, int n])
```

Extracts *n* characters from text string *t* starting at character position *s*. If *n* is omitted, all characters from text *t* up to the end of the string will be extracted. Either or both *s* and *n* may be negative to count from the end of the string.

e.g. `Text("abcdefghij",4,3)` ➔ `"def"`

`Text("abcdefghij",4)` ➔ `"defghij"`

`Text("abcdefghij",-4,3)` ➔ `"ghi"`

`Text("abcdefghij",4,-2)` ➔ `"defgh"`

`Text("abcdefghij",-4,-1)` ➔ `"ghi"`

Insert

```
text Insert(text t, text into, int start [, int length])
```

Inserts the text string *t* into the text string *into* at the character position given by *start*. The *start* parameter may be negative to count back from the end of the *into* string. If *length* is positive, that many characters are omitted from *into*. If *length* is negative, it counts back from end of string *into*.

e.g. `Insert("123","abcdefghij",5)` → `"abcd123efghij"`

`Insert("123","abcdefghij",5,2)` → `"abcd123ghij"`

`Insert("123","abcdefghij",5,-2)` → `"abcd123ij"`

Replace

```
text Replace(text t, text c, text r)
```

Replaces every occurrence of character *c* with *r* in text string *t*.

N.B.: Only the first character of texts *c* and *r* are used.

This is useful for replacing, say, commas with tabs, etc.

Trim

```
text Trim(text t [, text chars])
```

Removes any characters found in *chars* from both ends of the text string *t*. If *chars* is not defined, white-space characters are removed (space, tab, newline, carriage return, form feed).

e.g. `Trim("   abc def   ")` → `"abc def"`

LTrim

```
text LTrim(text t [, text chars])
```

As for **Trim**, but only removes characters from the start of the string.

e.g. `LTrim("   abc def   "," abf")` → `"c def   "`

RTrim

```
text RTrim(text t [, text chars])
```

As for **Trim**, but only removes characters from the end of the string.

AddSlashes

```
text AddSlashes(text t [, text chars [,text escStr]])
```

The text string *t* is processed so that certain characters, listed in *chars,* are converted to another form introduced by an escape character given in *escStr*. By default the escape character is a backslash \, and the characters to be converted are newline, carriage return,

tab and double-quotes. The newline, carriage return and tab characters are converted to the conventional form described in section 16.4. Any other control characters listed in *chars* will be converted to hexadecimal form preceded by the escape character and `x`. Other characters are simply preceded by the escape character. The escape character will itself always be escaped if it occurs in the string *t*, whether or not it appears in *chars*. If *chars* is specified as an empty string, no characters are converted at all.

The string can be unescaped using the **StripSlashes** function. Note that escaping the lower case letters `n`, `r`, `t` or `x` will give rise to ambiguities when stripping slashes later.

For example,

```
list += ","+AddSlashes(item, ",");
```

would add an item to a comma-separated list, escaping any commas in the item itself with a backslash. This is useful if you wish to present a menu to the user using the **Menu** function.

StripSlashes

```
text StripSlashes(text t [, text escStr])
```

Unescapes characters in text *t*, thus reversing the action of the **AddSlashes** function. By default the escape character is a backslash \, but this can be set to a single character of your choice using the optional second parameter *escStr*.

UrlEncode

```
text UrlEncode(text t)
```

Converts the string so that it is suitable for use in a URL. All characters except letters, numbers, underscore and dash will be converted to %nn where nn is a two-digit hexadecimal code. This function is useful for encoding data as though it has been typed into a form on a web page and then submitted. See the street map and journey planner action scripts in Section 17.16.

Val

```
real Val(text t)
```

The numerical translation of text string *t*.

Format

```
text Format(real x, text f)
```

The textual version of number *x* formatted according to *f*. The format codes are those used in the C programming language, e.g. `%l.2f` to format a real number to two decimal places.

RomanNumber

```
text RomanNumber(int n)
```

Number *n* represented by upper case Roman numerals. If lower case numerals are

required, you can use the **LCase** function to convert the result. The number must be in the range from 1 to 3999.

Money

```
text Money(real x)
```

A text string representing the given amount (in sterling). This is useful for cheques and receipts, etc.

e.g. `Money(2.34)` ➔ `"Two Pounds 34 Pence"`

Char

```
text Char(int n)
```

The character represented by ASCII value *n*.

Ascii

```
int Ascii(text c)
```

The ASCII value of (the first character of) text *c*.

Sys

```
text Sys(text v)
```

The translation of the given RISC OS system variable *v*, If the variable is not defined, a blank string is returned.

16.11 Date functions

Where dates are entered with a two digit year, years less than 20 will be assumed to be 2000-2019, while values of 20 and higher will be assumed to be 1920 onwards.

Today

```
date Today()
```

Today's date-time code.

Secs

```
real Secs(date d)
```

The seconds for given date/time *d* to the nearest hundredth of a second.

N.B. This function returns a real number to two decimal places.

Mins

```
int Mins(date d)
```

The minutes for given date/time *d*.

Hrs

```
int Hrs(date d)
```

The hours for given date/time *d*.

Day

```
int Day(date d)
```

The day number for given date *d*.

e.g. `Day(Date("18 April 2009"))` ➔ `18`

Month

```
int Month(date d)
```

The month number for given date *d*, where January is numbered one, February two, and so on.

Year

```
int Year(date d)
```

The year in 4 digit form for given date *d*.

DaysInMonth

```
int DaysInMonth(date d)
```

Returns the number of days in the month containing the date *d*. For example, `DaysInMonth( Date("20/2/2000") )` will return 29.

Date

```
date Date(text t [, text f])
```

The date translation of the text *t*. If the date cannot be converted, or the date does not exist, an error will be generated. The optional *f* parameter specifies the format the date is to be expected in. If this is omitted, *day*/*month*/*year* is assumed. For example, to convert an ISO format date (*yyyymmdd*) you could use:

```
Date("20050220", "%ce%yr%mn%dy");
```

Impact will accept months in numbers or words wherever `%mn`, `%mo` or `%m3` is given. For a complete list of date format codes, see *Appendix 3: Date format codes*.

Age

```
real Age(date b [, date d])
```

The age of a person at date *d*, if born on date *b*. If *d* is not given, today's date is assumed. The result is in years, and may be fractional.

AddMonths

```
date AddMonths(date d, int m)
```

Return the date *m* months from *d*. The integer *m* may be negative. If the resulting month does not have enough days in it for the day to remain the same, the last day of the month will be used.

AddYears

```
date AddYears(date d, int y)
```

Return the date *y* years from *d*. The integer *y* may be negative. If the resulting month does not have enough days in it for the day to remain the same, the last day of the month will be used.

DFormat

```
text DFormat(date d, text f)
```

Returns the given date *d* in format *f*. Date formats are explained in *Appendix 3: Date format codes*.

TextMonth

```
text TextMonth(int m)
```

Returns the month number *m* in textual form.

TextDay

```
text TextDay(int d)
```

Returns the day in-the-week number *d* in textual form, where 1 represents Sunday.

16.12 Maths functions

In the event of an error in these functions, such as *x* being out of range for the function, an error will be displayed, and the result will be zero in all cases.

Trunc

```
real Trunc(real x, int n)
```

x truncated to *n* decimal places. The value *n* may be negative, as shown in the following examples.

```
e.g. Trunc( 123.456, 2 )  → 123.45

     Trunc( 123.456, 0 )  → 123

     Trunc( 123.456, -2 ) → 100
```

Round

```
real Round(real x, int n)
```

x rounded to nearest *n* places. The value *n* may be negative, as shown in the following examples.

e.g. `Round( 123.456, 2 )` ➔ `123.46`

`Round( 123.456, 0 )` ➔ `123`

`Round( 123.456, -2 )` ➔ `100`

Int

```
int Int(real x)
```

The integer value of *x*. `Int(x)` is equivalent to `Trunc(x,0)`

Fract

```
real Fract(real x)
```

The fractional part of *x*. This will always lie between -1 and 1. The sum of `Fract(x)` and `Int(x)` is equal to *x*.

Abs

```
real Abs(real x)
```

The absolute value of *x*.

Fact

```
real Fact(int n)
```

n!, the factorial of *n*.

Tan

```
real Tan(real x)
```

The tangent of angle *x* (*x* in radians).

Sin

```
real Sin(real x)
```

The sine of angle *x* (*x* in radians).

Cos

```
real Cos(real x)
```

The cosine of angle *x* (*x* in radians).

ATan

```
real ATan(real x)
```

The angle (in radians) whose tangent is *x*.

ASin

```
real ASin(real x)
```

The angle (in radians) whose sine is *x*.

ACos

```
real ACos(real x)
```

The angle (in radians) whose cosine is *x*.

Log

```
real Log(real x)
```

Log to the base *e* of *x*.

Log10

```
real Log10(real x)
```

Log to the base 10 of *x*.

Sqrt

```
real Sqrt(real x)
```

The square root of *x*.

Exp

```
real Exp(real x)
```

e raised to the power of *x*.

Hyp

```
real Hyp(real x, real y)
```

Hypotenuse of a right-angled triangle with sides *x* and *y*.

Rads

```
real Rads(real x)
```

Angle *x* (in degrees) converted to radians.

Degs

```
real Degs(real x)
```

Angle *x* (in radians) converted to degrees.

Pow

```
real Pow(real x, real y)
```

x raised to the power of *y*.

Pi

```
real Pi()
```

The value of π (3.1415926...)

Min

```
real Min(real a, real b, real c...)
```

The minimum value of *a*, *b*, *c*...

Max

```
real Max(real a, real b, real c...)
```

The maximum value of *a*, *b*, *c*...

BaseConvert

```
text BaseConvert(text x, int base-to [, int base-from])
```

Convert number *x* from base *base-from* to *base-to*.

The bases can range from 2 to 36. Letters A-Z are accepted for digits from 10 to 36. On input they may be in mixed case. Output is in upper case - if you want lower case, you can use the **LCase** function to convert it.

If *base-from* is not specified, base 10 is assumed.

Only integers may be converted. The number is considered terminated if a space or control character is reached. Spaces at the start of the input string are skipped.

Errors are raised if the bases are out of range, an out of range input character is found, or a decimal point is found, or on overflow.

```
e.g.  BaseConvert( "255", 16 )    → "FF"

      BaseConvert( 255, 8 )       → "377"

      BaseConvert( "FF", 10, 16 ) → "255"
```

16.13 Miscellaneous functions

Record

```
int Record([text n])
```

If *n* is the name of a database, returns the number of the current record selected in that database. If *n* is not given, the function returns the number of the current record in the

database from which the calculation was issued. Record numbers are assigned as positive numbers starting at 1 and are not changed throughout the existence of the record.

If *n* is a valid foreign key field name, returns the number of the record in the foreign database to which the field is referring. The foreign key field may be given in the form `database:field` if you wish to state the database explicitly: otherwise the current database is assumed.

Returns zero if the specified database or foreign field does not exist. If the specified database was not open when the function is evaluated, it will be opened in the process, but a zero value will be returned.

In common with all other calculation functions, only databases which are held in the same directory as the current database may be accessed. Thus a calculation expression in a database in *UserData* may not access databases in *SampleData*.

Next

```
int Next([text d])
```

Selects the next record (according to context) in database *d* (or the current database if *d* is omitted), and returns its number. A value of zero indicates that no more records are available.

Previous

```
int Previous([text d])
```

Selects the previous record (according to context) in database *d* (or the current database if *d* is omitted), and returns its number. A value of zero indicates that no more records are available.

Safe

```
int Safe([text d])
```

Returns non-zero (true) if it is safe to change records in database *d* (or the current database if *d* is omitted). The action depends on the user's settings in the **Choices** window. If there is any unsaved data on a card, it is either ignored, saved automatically, or the user is warned. Zero (false) is returned only if the user selects *Cancel* in the message window.

It is not permitted to use this function in the definition of a calculation field, the default value expression for a field, or in defining a calculation filter.

Modified

```
int Modified([text d])
```

Returns non-zero (true) if the record currently displayed in the card has been modified since it was last saved. If the database *d* is not specified, the current database is assumed. This function essentially reflects whether the database card is displaying an asterisk at the end of the window title.

Note that if the user is editing the contents of an existing file field, changes are not detected by Impact when the file is saved, so such modifications to file fields are not reflected in the window title. Dragging a new file into a file field is reflected in the modification state, however.

It is not permitted to use this function in the definition of a calculation field, the default value expression for a field, or in defining a calculation filter.

FieldValue

```
int|real|text|date FieldValue(text f)
```

This function returns the value of the field *f* in its internal format, i.e.

- the numeric value of a number, flag, option, menu or browser field
- the text of a text or sequence field
- the date or time for a date or time field
- the filename as text for an external file field
- the contents of an internal file if it is a text file, otherwise the name of the filetype.
- the result of an action or calculation field
- the caption of a label field

e.g. `FieldValue("Videos:type")`

For a function which returns the contents of a field in a textual form, see **FieldText**.

FileName

```
text FileName(text f [, int force])
```

The name of an internal or external (linked) file field must be passed to this function. The function will return the full pathname of the file referenced or contained in the current record, or a blank string if there is no file present.

The main use of this function is in making e-mail attachments with ImpEmail based on external or linked file content.

For internal file fields, it can sometimes be useful to know the filename that Impact would use to store the file, even if there is no file currently stored. Setting the optional second parameter to a true value will achieve this. It could be used in an action script to copy a file from elsewhere to the correct location.

CardFile

```
text CardFile(text f)
```

This function is similar to **FileName**. You must pass it the name of an internal or external (linked) file field. The function will return the full pathname of the file stored in the field, but unlike FileName will return a value which has not yet been saved. It is similar in concept to the **CardText** function. For external file fields they behave identically, but for internal file fields CardText returns the content of the file (if it is a text file) or the name of the filetype.

FileType

```
int FileType(text n)
```

This function takes as an argument a full pathname for a file or directory, and returns an integer indicating the type of the file. As well as the ordinary filetype numbers, which range from 0 to 4095 (0 to FFF in hexadecimal) there are various special codes which may be returned, as follows:

Type (decimal)	Type (hex)	Meaning
4096	1000	Directory
8192	2000	Application directory
12288 + filetype	3000 + filetype	Image filing system (e.g. 3A91 = Zip file)
16384	4000	Untyped file (load + exec address)
20480	5000	Incomplete file

An error will be generated if the file cannot be found.

FileSize

```
int FileSize(text n)
```

The argument should be the full pathname for a file. The size of the file in bytes is returned. An error will be generated if the file cannot be found, or if the name leads to a directory rather than a file.

You could use this in an action script attached to a file field to check that the size of the file falls within chosen limits:

```
if (FileSize(CardFile("Picture")) > 1024*1024) {
    Error "The Picture file must be 1MB or less in size!";
}
```

FileContent

```
text FileContent(text n [, int limit])
```

This function takes a full path to a file as its argument. The function loads the file and returns the content. In order to avoid problems, only the first 16K of the file is returned. You can stipulate a different limit using the optional second parameter, which sets the limit in bytes.

Note that with a binary file (i.e. one which is not simply text) then any content after the first null byte will not be returned. This is because Impact's expression engine uses the null byte to terminate string values.

An error will be generated if the file cannot be read or the name indicates a directory.

This function can be used in conjunction with the **FileName** function to read the content of a file linked to from a database record:

```
FileContent(FileName("Picture"))
```

Flag

```
int Flag(int f, int b)
```

This function can be used with the value of a flag field to determine whether an individual button is selected. In the sample *Addresses* database, there is a flag field called *Cardlist*. We can test whether the first (and only, as it happens) button in this field is set by obtaining the value `Flag(Addresses:Cardlist, 1)`

In general terms, the function returns a non-zero value, which will test as true, if button *b* is selected in flag field *f*. It is equivalent to `(f & 1<<(b-1))`, where `&` is the bitwise AND operator. The buttons are numbered from 1 to 32. The function may also be used with binary values other than flag fields.

Button

```
int Button()
```

The mouse button being pressed at the time the function is used.

Returned values are:-

0	none
1	Select
2	Menu
3	Adjust

LastButton

```
int LastButton()
```

The mouse button last pressed; values as for **Button**.

KeyDown

```
int KeyDown(int n)
```

Returns non-zero (true) if the given key number is currently pressed. Key numbers are as given in the RISC OS *Programmer's Reference Manual*.

ToolState

```
int ToolState(text t)
```

Returns non-zero (true) if the given tool is selected (shown depressed).

Index

```
text Index([text d])
```

The name of the currently selected index in database *d*. If *d* is not given, the current database is assumed. If there is no index currently selected, the empty string is returned.

e.g. `Index("Videos")`

Filter

```
text Filter([text d])
```

The name of the currently selected filter in database *d*. If *d* is not given, the current database is assumed. If there is no filter currently selected, the empty string is returned.

e.g. `Filter("Addresses")`

Field

```
text Field([text d])
```

The name of the currently selected field map in database *d*. If *d* is not given, the current database is assumed. If there is no field map currently selected, the empty string is returned.

Indexes

```
text Indexes([text d])
```

A comma separated list of index names defined for database *d*. If *d* is not given, the current database is assumed. This function is useful for creating an index menu.

Filters

```
text Filters([text d])
```

A comma separated list of filter names defined for database *d*. If *d* is not given, the current database is assumed. This function is useful for creating a filter menu. Filters which have been set to be hidden will not appear in the list.

Fields

```
text Fields([text d])
```

A comma separated list of defined field map names for database *d*. If *d* is not given, the current database is assumed. This function is useful for creating a field map menu.

Card

```
text Card()
```

The name of the card from which the calculation was issued. It is not permitted to use this function in the definition of a calculation field, the default value expression for a field, or in defining a calculation filter. It is intended for use in action scripts only.

CField

```
text CField()
```

The name of the field currently containing the caret, if any. This is in the form `database:field`

Exists

```
int Exists(text f)
```

Returns 1 (true) if the field *f* exists in the local database, 0 otherwise. To test for the existence of a field in another database, *f* may include the database name in the form `database:field`

e.g. `Exists("surname")`

`Exists("Videos:film")`

Valid

```
int Valid([int rec [, text d]])
```

Returns 1 (true) or 0 (false) according to whether the record numbered *rec* exists, passes the current filter test and appears in the current index of database *d*. If the database is omitted, the current one is assumed. If the record number is omitted, the current record is assumed.

IsNull

```
int IsNull(<expression>)
```

Returns 1 (true) if the expression yields a null result, 0 otherwise. *Null* is a value meaning the datum is essentially unknown (a text field which has been set to an empty string or a number field set to zero are not considered null). Null values currently can only appear in date fields.

e.g. `IsNull( Date("") )` → `1`

`IsNull( Date("") + 10)` → `1`

`IsNull( "" )` → `0`

Macro

```
text Macro(<macro>, <document>)
```

Ovation Pro only, the **Macro** function may only be used in action scripts after a merge connection has been established by the action command `MergeTo "OvationPro";`

Sends the given macro command to the named document in *Ovation Pro* for execution and returns the result, which must be a string. See also the **Macro** action command in Section 17.8.

The syntax of **<macro>** is that of a standard *Ovation Pro* macro command, e.g.

```
"{macv = itos( getunits() ) }"
```

See your *Ovation Pro* manual for more details.

Note that if the *Ovation Pro* function you wish to use returns an integer (as with **getunits**() above), you will need to convert the result to a string using the *Ovation Pro*

function **itos**().

Phone

```
text Phone()
```

The last received number from a caller-id unit (see *Chapter 20: Caller display*).

16.14 Conditional calculations

Calculations may include the following operators:

```
A ? B : C
```

The value of this expression depends on the value of *A*, such that if *A* is non-zero, the expression's value will be *B*, otherwise it will be *C*, where *A*, *B*, and *C* may be any valid sub-expressions.

For example, if *male* is a **flag field** containing a single tick box, and *birthdate* is the date of birth, then

```
(male ? 65 : 60) - Age(birthdate)
```

will calculate the number of years until retirement for either sex (assuming men retire at 65 and women at 60). This is because the expression `male ? 65 : 60` will have the value 65 if *male* is 1 (the flag is set) and 60 otherwise.

16.15 Using flag fields

When used in a calculation, flag fields yield a numeric value according to which flags buttons are selected. Each button in a single flag field is given a value, and where several buttons are selected the field value is the total of the selected button values.

The values assumed are as listed below:

Button	Value
1	1
2	2
3	4
4	8
5	16 etc.

For example, if buttons 1, 3 and 4 are selected, the field value will be 13.

These values are useful in the conditional calculation construct described in the previous section, to control the result. For example -

```
cost * (vat ? 1.175 : 1)
```

will compute a total including or excluding vat according to the setting of a flag field called *vat* (since *vat* will be 1 if set, and 0 otherwise).

Where there is more than one flag button in a set, the **Flag** function may be used to test if a particular button is set.

```
Flag(field,button)
```

This will return a non-zero result if the given button number (1-32) is set in the given flag field.

For example

```
Flag(types,4) ? "Type 4" : "None"
```

The Flag function is very useful for setting up filters on flag fields. See *4.3 Calculation filters* for more information. A textual form of the flag button selection is given by the **FieldText** function.

Merging flag fields

When merging flag fields, a textual representation is used rather than numerical. This text is formed by concatenating the legends from all selected flag buttons in the set with comma separators. This form is also shown in the **table display**.

16.16 Using option, menu and browser fields

Using an option, menu or browser fields in a calculation results in text, the value being taken from the legend of the selected option or the text of the menu or browser item. If the field has no value set, the text used in the calculation will be blank.

You can obtain the numerical form of the field, as stored internally in the database, using the **FieldValue** function.

16.17 Using file and image fields

Internal file fields (where the files are stored within the database) will return the content of the file, but only if the file is a plain text one.

For external file fields the filename is returned instead. It makes no difference whether the file can be found or not: the value is the full pathname of the file, as seen by the computer on which you are running Impact.

There are various functions which will help you get the name or content of the files more consistently, no matter whether the file is stored internally to the database or externally:

`FileName`	The file name of the material stored in the field.
`CardFile`	The file name for a file field which is not yet saved from the record card to disc.
`FileContent`	A function to read the content of any file (takes a filename, rather than a field name as its argument).

16.18 Operators

Operators are applied according to a priority rating. For example, multiplication is always done before addition, as addition has a lower priority. The following table gives all operators and their priorities.

Operator	Action	Priority
=	assignment	2
+=	add and assign	2
-=	subtract and assign	2
*=	multiply and assign	2
/=	divide and assign	2
%=	mod and assign	2
? :	conditional construct	3
||	logical OR	4
&&	logical AND	5
|	bitwise OR	6
^	bitwise XOR	7
&	bitwise AND	8
==	equal to	9
!=	not equal to	9
<	less than	10
>	greater than	10
<=	less than or equal to	10
>=	greater or equal to	10
>>	bit shift right	11
<<	bit shift left	11
+	add	12
-	subtract	12
*	multiply	13
/	divide	13
%	mod (remainder)	13
^	bitwise not	14
!	logical not	14

Note that it is not permitted to use assignment operators (=, +=, -=, *=, /=, %=) in the definition of a calculation field, in the default value expression for a field, or in defining a calculation filter. They are intended for use in action scripts only.

Chapter 17: Action commands

This chapter describes the action commands available, and gives examples of their use. How to place an action button on a card, and how to create action scripts, is described in Chapters 12-14.

Action buttons are a powerful feature of Impact allowing many routine actions to be carried out with a single key press or mouse click, such as printing a label or report, entering specific data into fields, or displaying a table of selected information.

The action commands and calculation functions form a programming language for controlling Impact. As with any programming language, it is easier to learn by modifying existing scripts, so some examples are provided in *17.16 Examples*. There are also several working examples provided in Impact's sample databases, and it is well worth exploring these for ideas. To see the script for an action button, just click with ***Adjust*** on the button.

17.1 Execution context

There is a subtle difference between actions carried out as the result of a tool on the toolbar being used, and those carried out when an **action button** is clicked.

When an action button is clicked, execution is carried out within a separate **context**. This context inherits all existing settings of filters, indexes, field maps and the database record number when first started. However, all changes to these made within the action script only apply during its execution, and are discarded when it terminates.

However, this is *not* the case when a tool action script is executed, allowing these to make permanent changes to the current record number and the filter, index and field map selection.

17.2 Action command syntax

Action commands consist of a **keyword** defining the action, followed by one or more names or other data, as described below.

In the following descriptions items within `<angle brackets>` should be replaced with the required item, while items shown within `[square brackets]` are optional and may be omitted. All other characters should be included as shown, particularly commas.

Important: All commands should be terminated with a semi-colon. Also, all action commands are *case insensitive*. However, this does **not** apply to field names, variables (see below) or functions, which are case sensitive.

All data items following the keyword will be executed as **expressions**, and may therefore contain references to fields. As with other calculations, this means that literal text must be given between double quotes, to distinguish them from a field name. Expressions are fully described in *Chapter 16: Calculations*.

Many commands may optionally include the name of a database. If this is not given, the database to which the action button belongs will be assumed, so that in most cases it may be omitted. Note that if supplied, the database name **must** be terminated with a colon. Named databases are opened automatically, but their cards are not displayed unless requested by an action command.

17.3 Field assignments

```
[>][<database>:]<field> = <expression>;
```

If a keyword is not recognised, Impact will attempt to interpret it as a **field name**, and assign the evaluated expression to the field. In cases where `<database>` or `<field>` may clash with a keyword, using the optional `>` character at the start of the command will force the command to

be interpreted as an assignment.

The expression is executed and assigned to the field specified in the current record. When this is done, the appropriate card(s) will be flagged as modified, requiring the **Save** command (or a click on the **save tool** in the relevant database) to save the modified data to disc. In effect assignment only modifies the contents of the card on view.

Assigning to flag, option, menu and browser fields

When assigning to these types of fields, the expression may yield either a number or a text string. If a number is produced, the value will be assigned directly to the field. For example, assigning the value 2 to a menu field will cause the second menu text item to be shown in the field.

If the expression produces a text string value, this will be matched against the captions of the field options or the menu texts, and the appropriate item will be selected and shown on the card. In the case of flag fields, multiple values may be selected by listing the text captions of all those required separated by commas. All the other flags will be deselected.

Assigning to date fields

Some expressions may yield a date field directly, for example

```
AddMonths(ExpiryDate,4)
```

but other may require conversion from a text string. If so, the format of the date in the text string must match the format used on the card for the display of the field. Suppose you had a date field called *ExpiryDate* and a number field called *Year* which was set equal to 1990. It is no good trying to assign

```
ExpiryDate = "4 May "+Year;
```

if your date field is formatted as %ce%yr:%mn:%dy and would show the above date as 1990:05:04. To get round this situation, you may need to convert the string to a date explicitly by using the **Date** function with an appropriate format:

```
ExpiryDate = Date("4 May "+Year, "%dy %mo %ce%yr");
```

Assigning to foreign fields

Dependent fields may be assigned new values by whatever means is appropriate for the underlying data type. Assignment to a **key field**, however, expects an expression giving the number of the record in the foreign database to which the key field should be linked.

Field assignment examples

The following are examples of action commands which might be used in a *Videos* database, linked to a customer *Addresses* database.

```
dueback = Today() + hireperiod;
```

assigns today's date plus the number of days indicated in field *hireperiod* to a local date field called *dueback*, indicating the date the video is due back.

```
count = count + 1;
```

increments the value in field *count* by 1.

```
Addresses:hired = Addresses:hired + 1;
```

increments the number of videos in the *Addresses* database which the customer currently has on hire.

```
name = Record("Addresses");
```

Where *name* is the local **key field** related to the *Addresses* database, this command assigns the number of the record currently displayed in the *Addresses* database to the key field. This has the effect of inserting the foreign name and address as though it had been dragged across.

It is possible to access data from the related record in the foreign database, rather than from the currently displayed record. This is done using the key field in the local database which holds the relational link, as explained in Chapter 16: *Calculations*.

```
total = name.hired;
```

A local field *total* showing the number of videos hired by the customer is updated directly from the *Addresses* database field *hired*, by linking through the local key field *name* to find the appropriate customer record in the *Addresses* database.

```
Videos:total = Videos:name.hired;
```

This does the same as the previous example, but may be done from a third unrelated database as it has explicitly included the *Videos* database name. Note that using the key field syntax in this manner it is not necessary to know the name of the *Addresses* database as this is implicit in the key field definition.

Note that though it is possible to **read** the value of the field *hired* in *Addresses* by referring to it as `Videos:name.hired`, it is not possible to **assign** a value to the field *hired* using this expression.

17.4 Star commands

Action scripts may include RISC OS star commands such as `*copy`, `*echo`, etc.

Multiple expressions enclosed in parentheses may be evaluated in the star command. For example, suppose the *phone* field equalled 345181. The following star command in an action script

```
*echo (phone, "x", 45*2) is the ("correct"+"number;");
```

would be equivalent to

```
*echo 345181x90 is the correct number;
```

Star commands are terminated by a `;` or `!` which is not within double quotes or within parentheses, or else by the end of the line.

Note that star commands should not include the character `!` as this is interpreted as the start of a comment (see below). If you need to include `!` in your command, put it in double quotes within parentheses, e.g.

```
*wipe ("!ReadMe");
```

which will be interpreted as

```
*wipe !ReadMe
```

A further example is given in the *Keeping a record* script at the end of this chapter.

17.5 Comments

The characters ! | and # may all be used to indicate the start of a comment, whether at the start or part-way through a line. If you wish to place a comment on the same line as a star command, only ! may be used.

17.6 Indexes, filters, and field maps

Index

```
Index [<database>:] [<name>];
```

Select the named index, which will be active until changed or until the end of the action script. If <name> is omitted, any current setting will be cancelled.

Examples:

```
Index "BySurname";          ! Sort by name in current database

Index Videos:"DateHired";   ! Sort videos by hire date

Index Videos:;              ! Clear index for Vidoes database
```

Filter

```
Filter [<database>:] [<name>];
```

Select the named filter, which will be active until changed or until the end of the action script.

If <name> is omitted, any current setting will be cancelled.

Fields

```
Fields [<database>:] [<name>];
```

Select the named field map, which will be active until changed or until the end of the action script.

If <name> is omitted, any current setting will be cancelled.

TempFilter

```
TempFilter [<database>:] "<calculation>";
```

Create a temporary calculation filter. Only one temporary filter can be defined at any one time, and will be destroyed when the action script terminates. Such a filter will take precedence over any other filter which may be active.

e.g. `TempFilter Videos: "hired && Today()-hired > 7";`

TempFields

```
TempFields [<database>:] "<field-list>"[, "<cat-list>" [, "<cat-string>"]];
```

Create a temporary field map containing the fields listed in <field-list>, with those fields listed in <cat-list> concatenated by <cat-string> if given. Only one may be defined at any one time, and will be destroyed when the script terminates. The field map created will take precedence over any other which may be active.

Example:

```
TempFields "title,name,address,postcode","name,postcode"," ";
```

This example defines a field map which produces two columns of output, the first being the title and name fields concatenated and the second being the address and postcode fields concatenated. The field values are concatenated using a single space between them.

17.7 Record manipulation commands

Record

```
Record [<database>:] <record>;
```

Select specified numbered record as the current record for the current or given database. The record data will not be displayed in the card: to display the data you should use the **Display** command instead.

First

```
First [<database>:];
```

Select the first record in the current context for the current or given database.

e.g. `First Videos:;`

```
First;
```

Last

```
Last [<database>:];
```

Select the last record in the current context for the current or given database.

Next

```
Next [<database>:];
```

Select the next record in the current context for the current or given database.

Previous

```
Previous [<database>:];
```

Select the previous record in the current context for the current or given database.

Find

```
Find [<database>:] <field>, <data> [, <match>];
```

Search the given field for the given data, from the current record. If a matching record is found, this record becomes the current one, otherwise there is no change.

If `<match>` is given and is *true*, case-sensitive matching is used.

Examples:

```
Find "Name", "smith";
Find Addresses:"Name", "smith";
Find Addresses:"Name", "Smith", 1;  ! Case-sensitive search

Local "field", "term";
field = AskText("Enter name of field to search:");
term = AskText("Enter search term:");
Find field, term;
```

Delete

```
Delete [<database>:] <record>;
```

Delete the given record number.

e.g. `Delete Addresses: 23;`

Save

```
Save [<database>:] [<record>];
```

Save all changed data to disc in the given or current record. If the database is in **entry mode**, then the record number will be ignored and the data saved in the next available record.

Display

```
Display [<database>:] [<record>];
```

Display specified record (or current record if not specified) in all open cards in the current or given database.

Scan

```
Scan [<database>:];
```

Scan records in a database starting with the current one (using the index/filter/field map currently set), in order to be able to use table calculation functions, in a similar way to the **Scan** merge command. To scan from the start of the database, you will need to first issue the command **First;** to select the first record in the database. For example, if you had a database of cash transactions, with a field named *amount* you could obtain the total as follows:

```
TempFields "amount";
First;
Scan;
```

```
Message "Total amounts: "+Total(1);
```

17.8 Document merging and printing

These commands allow complete control over document merging, and should be used in preference to the old *Merge* and *SendLetter* commands. Examples of their use can be found in the example scripts *Printing with Ovation Pro* and *Keeping a record* at the end of this chapter.

MergeTo

```
MergeTo <application>;
```

Select the application to merge to.

e.g. `MergeTo "OvationPro";`

Printer

```
Printer <name>;
```

Select the named printer. The name should be in the form which appears under the printer's icon on the icon bar.

e.g. `Printer "BJC-210";`

DocLoad

```
DocLoad <document>;
```

Load the file named `<document>` into the application. If just a filename is given, the document is loaded from within the database's Documents directory. Or a full pathname can be given to load it from elsewhere.

DocMerge

```
DocMerge <document>;
```

Tell the document application to merge data. The values of the merge fields in the document will reflect the fields in the current record.

e.g. `DocMerge "Invoice";`

DocFix

```
DocFix <document>;
```

Fix the merged data permanently.

DocPrint

```
DocPrint <document>;
```

Print the document in its current state (or in the case of merging to *ImpEmail*, send the e-mail message).

DocClear

```
DocClear <document>;
```

Clear any merged data (unless it's been fixed).

Macro

```
Macro <macro>, <document>;
```

Ovation Pro only. See also the **Macro** function in Section 16.13.

Send the given macro command to the named document in *Ovation Pro* for execution. The syntax of **<macro>** is that of a standard *Ovation Pro* macro command, e.g.

```
"{filescrap(currentfile())}"
```

See your *Ovation Pro* manual for more details.

Other examples of macro commands are included in the example scripts *Printing with Ovation Pro* and *Keeping a record* at the end of this chapter.

17.9 Label printing

LabQueue

```
LabQueue [<database>:] <label-name> [, <copies> [, <multiple>]];
```

Queue labels for printing later. The text queued for printing will depend on the current field map chosen. If the `<multiple>` parameter is set, all labels from the current one to the end are queued, otherwise just the current record is queued. To queue all labels in the database, you will need to first issue the command **First;** to select the first record in the database. If `<copies>` is zero or negative, one copy of each label will be printed.

The label name parameter should be a string matching the name of one of the label designs in *LabPrint*. It does not have to match the complete name: the beginning of the name will do, but the case of the letters must match precisely.

e.g. `LabQueue "L7l68";`

LabPrint

```
LabPrint [<copies [, <first-label> [, <auto-shrink>]]];
```

Print the current label queue. The optional parameters correspond to the *Copies of each label*, *Next label to use* and *Shrink text to fit label* settings from the label printing window. The default is to print the queue once, from label number 1, with the text **not** being shrunk to fit the label if it is too large. If `<copies>` is zero or negative, the queue will be printed once.

LabStyle

```
LabStyle <label-name>;
```

This command may be used prior to displaying the label tool box, to pre-insert the

preferred label style into the box.

KillQueue

```
KillQueue;
```

Cancel any queued labels.

17.10 Conditional commands

If

```
If ( <expression> ) <command>; | <command-block>
```

where `<expression>` is any valid calculation resulting in a numerical value, and `<command>` is any other valid action command. Note the (brackets) are required. If the result of `<expression>` is non-zero (true) the following `<command>` or `<command-block>` is executed, otherwise it is skipped.

Else

```
Else <command>; | <command-block>
```

If used, this *must* follow an **If** command, and will result in the following commands being executed if the expression was false.

While

```
While ( <expression> ) <command-block>
```

If `<expression>` is *true* (non-zero) the following commands will be repeatedly executed until the `<expression>` becomes false. This means that some item within the expression must be changed by the following commands, or execution will continue forever! If this should happen, the ***Escape*** key may be used to halt it.

<command-block>

Commands such as **If**, **Else** and **While** (see above) may be followed by a block of commands within curly brackets.

```
if ( a == b ) {
        command 1;
        command 2;
}
```

17.11 Displaying dialogue boxes

Various dialogue boxes may be opened from a command script.

Dbox

```
Dbox <name>, <state>;
```

This command opens or closes one of a number of dialogue boxes available within the Impact, and is principally used with tool action files. The boxes are:

index	The index definition box.
filter	The filter definition box.
fields	The field map definition box.
search	The search box.
merge	The document merge control box.
import	The CSV import box.
export	The export control box.
backup	The backup control window.

17.12 Card commands

Show

```
Show [<database>:] <card>;
```

Displays the given card on screen.

```
e.g.  Show Addresses: "Actions";
```

Close

```
Close [<database>:] [<card>];
```

Closes the given card. If the card name is omitted, the database will be closed.

```
e.g.  Close "Actions";     ! close card named 'Actions'

      Close Addresses:;    ! close Addresses database

      Close;               ! close current database
```

Clear

```
Clear [<database>:] [<card>];
```

Clears the named card, or all cards for the current database. This is useful prior to entering new data and using the **Copy** command.

Copy

```
Copy [<database>:] [<card>];
```

Saves the currently displayed data to a new database record.

Entry

```
Entry [<database>:] <state>;
```

Sets **entry mode** for given database, according to `<state>`, which should be *true* or *false*.

Select

```
Select [<database>:] <card>, <state>;
```

Sets given card into **select mode**, allowing fields to be selected in groups for drag exporting, etc.

17.13 Creating variables

User defined variables may be created using these commands. Note that, unconventionally, the variable names may be given as expressions.

A variable may represent any data type, numeric, text or date, depending on the value assigned to it, and does not have any intrinsic type.

Note that the assignment

```
fred = 12.6.97
```

will not assign a date to fred, as Impact cannot recognise free format dates as dates, and is likely to assume a numerical value. The **Date** function is correctly used in this instance:

```
fred = Date( "12.6.97" )
```

Local

```
Local <name> [, <name>...];
```

Creates a local variable called `<name>`,

e.g. `Local "fred", "bill";`

Local variables are only accessible within the command file in which they are defined, and take precedence over any global variables or built-in functions of the same name.

Attempts to create a local variable with the same name as an existing one will cause an error message.

Global

```
Global <name> [, <name>...];
```

Creates a global variable with the given name,

e.g. `Global "fred", "bill";`

Global variables may be used in calculations elsewhere in the program, in other databases and in action command files. They take precedence over any built-in functions of the same name, and are preserved from one Impact session to another, even if the machine is switched off.

Duplicate definitions of a global variable are ignored without error.

Note that normal database fields take precedence over all variables and built-in functions, where the names are the same. We recommend using all lower case for fields and variables, to avoid conflict with current or future built-in functions.

Global variable storage

Global variables are stored between program runs in Impact's *Choices* directory, in a file called *Globals*. This file is editable, which is the only current way to remove unwanted variables. The file format is:

<name>, <type>, <value>

where **<type>** may be *Number*, *Date* or *Text*. For example:

```
fred,   Text,   "This is a text string"
vat,    Number, 17.5
used,   Date,   1.6.97
```

17.14 Miscellaneous commands

Abort

```
Abort;
```

Aborts the execution of the current action command file.

Break

```
Break;
```

Causes rest of current command block to be skipped. In a **While** loop, this command will also terminate the loop.

Beep

```
Beep;
```

Causes the computer to beep!

Error

```
Error <expression>;
```

This causes a standard error box to be displayed showing the given text. Execution of the action file is aborted. `<expression>` may be any valid expression.

Export

```
Export [database:] <filename> [,<multiple> [,<style>]]
```

The records of the database are exported to a text file, according to the index, filter and field map in force at the time. This action command provides a way to automate the type of exports possible using the **Export tool** (see *18.2 Using the export tool*).

If the `<multiple>` parameter is not zero, all records from the current one to the end are exported, otherwise just the current record is exported. To export all labels in the database, you will need to first issue the command **First;** to select the first record in the database.

The optional `<style>` parameter should be a string matching the name of one of the pre-saved sets of export options. It does not have to match the complete name: the beginning

of the name will do, or even part of the name, and capitalisation is ignored. If `<style>` is not provided then the default style, as determined by the application choices, will be used. See *21.10 Default export format* for details.

```
e.g.  Export "ADFS::HardDisc4.$.records";
      Export Addresses:"ADFS::HardDisc4.$.address-export";

      First;
      Export "ADFS::HardDisc4.$.records", 1, "Excel";
```

HourGlass

```
HourGlass <state>;
```

Turns the hourglass on or off during long operations.

Message

```
Message <expression>;
```

This causes a standard Wimp message box to be displayed, showing the given text. Execution of the action file continues when *OK* is clicked. `<expression>` may be any valid expression. Messages longer than 251 characters will be truncated.

Modem

```
Modem <string>;
```

This command sends the given text string to a modem connected via the computer's serial port. This permits such things as telephone dialling, or even data transfer. Consult your modem manual for details of commands supported. Internal modems and podule serial ports are supported via the **SerialDev** and **SerialSend** commands.

Pause

```
Pause <time>;
```

Causes execution to pause for the given number of centiseconds.

Return

```
Return <expression>;
```

Defines a value for the action button (see *17.15 Using action fields as functions*). The action script is aborted when this command is executed. `<expression>` may be any valid expression.

SerialDev

```
SerialDev <name>;
```

Loads and initialises the serial block driver given by `<name>`. The *!SerialDev* application is required to have been 'seen', so that the system variable *SerialDev$Path* has been defined.

This enables instructions to be sent using the **SerialSend** command to modems and other devices connected to serial card podules, etc.

SerialSend

```
SerialSend <string> [, <port>];
```

Sends the given string to the currently loaded serial block driver (which must have been loaded using the **SerialDev** command). The command is equivalent to the **Modem** command, but allows devices other than the internal serial port to be used. The optional second parameter is an integer specifying the port number, which may be needed for some serial interfaces.

Table

```
Table [<database>:] [<title>];
```

Causes a table to be displayed, as defined by the current context. The optional title parameter allows you to set the text shown in the title bar of the table window.

Tool

```
Tool [<database:] <name>, <state>[, <exclusive>];
```

This command may be used to control the tool on the toolbar for the current or given database. If `<state>` is *true*, the tool button will be depressed, if *false*, it will be shown in its normal state. This command is principally used within tool action files. If the final optional parameter, `<exclusive>`, is *true* then the tool button will be reset to its normal state on all cards in all other databases. For example, this is used in the Backup tool script, because the Backup dialogue may only be open for one database at a time.

Trace

```
Trace <expression>;
```

Causes the expression to appear in a throwback window. This can be useful for tracing bugs when developing action scripts. `<expression>` may be any valid expression.

WimpPoll

```
WimpPoll [<time>];
```

Embedded in a **While** loop, allows multi-tasking while the loop is executing.

WARNING: Avoid doing anything which may upset the commands being executed, like closing the database!

If `<time>` is given, this command will also pause the current action for an interval up to the number of centiseconds given.

Url

```
Url <string>;
```

This command provides a means of sending e-mail or accessing web sites from within

Impact by launching a URL.

e.g.
```
Url "http://www.google.com/";

Url "mailto:someone@somewhere.com";

Local "u";
u = "mailto:" + email;
u += "?Subject=" + UrlEncode("Message from Impact");
Url u;
```

Note that the workable length of a URL launched in this way depends on the receiving application and the method of dispatch. For best results please ensure that the AcornURI module is installed on your computer. You can check to see whether this is activated by pressing ***Ctrl-F12*** and typing "help acornuri".

Several examples of the use of this command are found in *17.16 Examples*.

17.15 Using action fields as functions

An action button's field name may be used like any other to represent a value. The value used is defined using the **Return** command within the action script. This facility allows very complex calculations to be performed, and the results used in merged reports, other calculations, and even other action scripts.

17.16 Examples

It is often easier to learn a new programming language by modifying existing scripts, so a number of examples are provided here. There is also a script in *15.6 Entering foreign data* relating to editing foreign fields. For further ideas, it is worth studying the action scripts for the various buttons in Impact's sample databases.

Opening a table

An action button which opens a table display, with appopriate field map, filter and index applied can be a real time-saver, compared with manually applying the settings and opening the table. For example, in a database of the members of a society, you may wish to list those whose membership has expired, in surname order, with a certain selection of fields. The following action script can serve as a model for this:

```
Index "BySurname";           ! Select surname index
Fields "BriefDetails";       ! Select field map
Filter "Expired";            ! Select filter to apply
Table "Membership expired";  ! Open table showing suitable title
```

This does not change the choice of index, filter or field map selected on the record card, but changes them in the context of the action script only, so that these are in force when the table is opened by the action script.

You could omit the setting of the field map, for example, in which case the table opened by the action script would use the field map which was in force on the record card.

The field selection does not have to be applied using a field map which as already been defined: a temporary field map may be created and applied using the TempFields command. The TempFilter command provides the equivalent for filtering.

Here is an example from the Addresses database which lists all the videos on loan to the person shown in the current record:

```
! Display table of videos on hire to current person.

! Select videos whose "name" field points to the currect
! record in Addresses.
TempFilter Videos: "Record(\"name\") == " + Record("Addresses");
Index  Videos:"DateHired";                  ! Select date order.
Fields Videos:"HireList";                   ! Select fields.
Table  Videos:"Videos hired to " + Name;    ! Display table.
```

Note the use of the database name with the action script commands. The action button resides on a card in the Addresses database, so we need to name the Videos database explicitly to get the commands to take effect in that database.

The TempFilter command stores a string defining a temporary calculation filter which will be tied to the table. The Record("name") part is in a string (hence the backslashes to get the double quotes into the string) and so will be evaluated every time the filter is used, but the Record("Addresses") is outside the string, and so is evaluated only when the action script is run and the temporary filter is first created. If you were on the first record of the Addresses database, the temporary filter would be stored as

```
Record("name") == 1
```

The table stays with the original person no matter what record you have since moved to in the address database. The table will also refresh when further videos are borrowed or returned.

Composing an e-mail

In a database of names and addresses it is useful to have a button which will open your e-mail software and start composing an e-mail to the person shown on the record card. You can see the following script in use in the demonstration Addresses database.

```
! Action script to invoke an e-mail send.
! A suitable e-mail client must have been "seen".
! i.e. Pluto, Messenger, etc...
! The field containing the e-mail address is named "email".
! If your e-mail address field has a different name, modify the
! script accordingly.

if ( email != "" ) url "mailto:"+email;
else error "There is no e-mail address.";
```

Depending on your e-mail client, you may be able to set the subject of the message and even the text of the body of the message by using a variation on the ordinary mailto URL. All you might then have to do is check the message, make any small adjustments, and hit the *Send* button.

The mailto: URL scheme can support various e-mail header fields, including To, CC, Subject and Body, but which ones are available depends on your e-mail client. We have tested Messenger Pro and have succeeded in sending quite lengthy texts (over 70K) through from Impact as the body of the message.

The following script takes an address from a database field called ‘email’, and the body of the message comes from an internal file field called ‘message’. The script uses the UrlEncode function to process the body and subject line correctly for including in the mailto URL. Note that the first e-mail field is introduced by ‘?’ and subsequent fields are introduced with the ‘&’. This works in the same way as submitting forms on web pages.

```
! Compose e-mail with a subject line and body text specified.
Local "u";     ! Set up a variable to build the URL
if ( email == "" ) error "There is no e-mail address.";
u = "mailto:"+Trim(email);
u += "?Subject=" + UrlEncode("E-mail from Impact");
u += "&Body=" + UrlEncode(message);
url u;
```

Another example, for composing a single e-mail to a group of people, is given later in this section.

A more straightforward way to send e-mails from Impact is, of course, to use *ImpEmail* which is described in *Chapter 10: Mail merging and formatted reports* and *Appendix 4: ImpEmail manual*, but using 'mailto:' URLs is very handy for one-off messages where you do not mind editing and sending them off manually.

Looking up a street map

The next script example takes the postcode from a field in the database and constructs a URL which will bring up a map of the area using the web-site www.streetmap.co.uk. Note that while this script worked at the time of writing, April 2011, the format of URLs accepted by web sites is liable to change without notice. The field in this case is named 'PostCode' but you can easily adjust the script to suit your own database.

```
! Look up street map from postcode field.
Local "u";

if (PostCode == "")
Error "No postcode available";

u  = "http://www.streetmap.co.uk/ids.srf?mapp=idmap";
u += "&searchp=ids&name=";
u += UrlEncode(PostCode);
Url u;
```

Note the use of the UrlEncode function which converts the data into a form suitable for submitting in the URL.

Looking up travel information

A more complicated example uses the postcode in a record card to plan a UK journey using the www.transportdirect.info car and public transport journey planner. The user will be asked which day they intend to travel, and at what time they wish to depart. You would need to modify this script to set the starting point for the journey to your own postcode: the example below starts from central Oxford.

```
! Look up travel planning information

Local "u";
Local "origin";
Local "d", "dd";

! Postcode to start the journey at
origin = "OX1 4AW";
```

```
if (PostCode == "")
Error "No postcode available";

! Ask user for date of journey
dd = DFormat( Today(), "%ZDY/%ZMN/%CE%YR" );
d = AskDate("Enter date on which you are travelling:", dd);

! Ask user for desired departure time
Local "t";
t= Menu("Departure time", "0800,0900,1000,1100,1200,1300,1400");
if (t=="") Abort;         !Abort if user chooses nothing

! Construct URL for www.transportdirect.info
! See http://www.transportdirect.info/Web2/ ...
! ... downloads/BusinessLinksTechnicalGuide.pdf

u  = "http://www.transportdirect.info/web2/journeyplanning/";
u += "jpLandingPage.aspx?";
u += "et=jp";
u += "&oo=p";
u += "&o="+UrlEncode(origin);
u += "&do=p";
u += "&d="+UrlEncode(PostCode);
u += "&dt="+DFormat(d, "%DY%MN%CE%YR");
u += "&t="+t;
u += "&da=d";
u += "&p=1";

Url u;
```

Scanning through a database

This example shows how to loop through a database, and perform some action on each record. In this case to find the total sales and vat values for a VAT return.

The filter *VatPeriod* is used to restrict action to records having a date within the current VAT period.

The final totals could be merged into a report printed within this script, or the variables made global, and used externally.

```
Local "a", "b", "c";
Filter "VatPeriod";     ! Use current period sales.
First;                  ! Select first record.
a = totcost;            ! Initialise totals.
b = totvat;
c = total;
```

```
While ( Next() )        ! Loop through database.
{
   a += totcost;        ! Add field values
   b += totvat;         !  if within VAT period.
   c += total;
}

! Tell user the total values.
Message "Sales = "+a+", VAT = "+b+", Total = "+c;
```

E-mailing a group of people

By looping through the database, as in the previous example, we can produce a script to send a single e-mail to a group of people. The idea is that you would apply a filter to identify the records, and then click the action button, which might be labelled "E-mail all". The script also reports to the user the number of records where no e-mail address was found, so that the user is alerted to make contact with these people by other means.

```
Local "addresses", "n";

First;                          ! Move to first record

if (email != "")  addresses += ","+email;
else              n += 1;

While ( Next() )
  if (email != "")  addresses += ","+email;
  else              n += 1;

addresses = Text(addresses,2);   ! Remove comma from start

if (n > 0)  Message n+" records did not have e-mail addresses."

if (addresses !="")  Url "mailto:"+addresses;
```

Since the release of *ImpEmail* it is generally more satisfactory to create a message using *ImpEmail* and generate messages to multiple people using Impact's ordinary mail merge facilities. *ImpEmail* is most frequently used to send individual e-mails to individual people, rather than one message to a group all at once, but by merging a table into the To field it is possible to send to a group quite easily. Nonetheless, you may have a need to use a script to assemble the list of e-mail addresses. As *ImpEmail* does not respond to the `Url` command, the script would need rewriting using a slightly different approach.

Let us assume we are using the sample *Addresses* database. First we create an *ImpEmail* document stored inside the documents area for the database:

1. Open the **Merge tool** and click the menu icon next to the Document box.
2. Select '(New)' from the Documents menu. The 'Create new merge document' window appears.
3. Make sure *ImpEmail* is the chosen application and enter the name for the document (we will assume it is called 'Test') and click *Create*.

A new *ImpEmail* document window should appear. At this point we can type any parts of the message which might remain constant, like the From address, the subject, perhaps, and the body of the message with a signature.

To be able to send the message to a group of people at once, dynamically identified by the action script, we need a merge command for *ImpEmail* to send to Impact to fetch them. the easiest way to do it is to use a calculation expression with a local variable. In the 'To' box, enter the following text:

```
{merge ":Impact.Addresses Calc EmailAddrList"}
```

The variable name we have chosen, 'EmailAddrList', can be changed to anything you want, but needs to correspond to the name used in the action script. The name of the database, 'Addresses', might need to be changed if you were using this technique with your own database.

The action script above then translates to the following:

```
Local "EmailAddrList", "n";

First;                              ! Move to first record

if (email != "")  EmailAddrList += ","+email;
else              n += 1;

While ( Next() )
  if (email != "")  EmailAddrList += ","+email;
  else              n += 1;

EmailAddrList = Text(EmailAddrList,2); ! Remove comma from start

if (n > 0)  Message n+" records did not have e-mail addresses."

if (EmailAddrList !="") {
    ! Change next line according to name of the document file...
    DocLoad "Test";
    MergeTo "ImpEmail";
    DocMerge "Test";
    DocFix "Test";
}
```

The `DocFix` command at the end fixes the merged elements into the document and makes the message editable, so you can edit it further before sending. Alternatively, the `DocPrint` command could have been used instead, to e-mail the message off without further editing.

This general technique, of building more complex content in a local variable and fetching it into a merge document can be very powerful. It is equally valuable when used with *Impression* and *Ovation Pro*.

Printing labels

It is often useful to have a button to queue an address label for printing later. This simple script performs this action.

A temporary field map is created to specify the fields required, and in this case a title field (Mr,

Mrs, etc.) is to be printed preceding the name field on the same line, separated by a space.

```
TempFields "title,name,address,postcode","name"," ";
LabQueue "L7l60";
```

Printing with *Ovation Pro*

This action script prints a merged document, assumed to be stored in the *Documents* directory within the database.

Using *Ovation Pro* has the advantage that a document may be closed after use, using the **Macro** command to send a macro to *Ovation Pro*. (See *Ovation Pro'*s manual for details of the *filescrap* command, etc.)

N.B.: Automatically closing the document afterwards cannot be done via *Impression*, which always leaves the document on screen to be closed manually, but the rest of the script would work the same way.

```
! Script to print an invoice.

Printer  "LaserJet 6";  ! Select printer.
MergeTo  "OvationPro";  ! Specify OvPro.
DocLoad  "Invoice";     ! Load document.
DocMerge "Invoice";     ! Merge document.
DocPrint "Invoice";     ! Print it.
Macro "{filescrap(currentfile())}", "Invoice";
```

Keeping a record

The next script is executed from the *Addresses* database. It copies a master letter into a letters' directory, merges and fixes the merged data, then saves the merged version. The event is logged as a new record in the *Letters* database, and a reference code generated by the **sequence field** *ref* used as the filename.

This method keeps a permanent copy of all letters sent, and a record in the *Letters* database. An action button could be included in this database to load and display the letter using the reference field from the card.

Note in this example how the star command ***Copy** is used to make a copy of a master template document file. If the parameters to any star command are enclosed in brackets, they will be evaluated as expressions within Impact, before being passed to the operating system.

Note also the use of the **Query** function to ask the user if the document is to be further edited after merging the name and address information from the database. If the user clicks OK, the script is aborted at that point, otherwise the document is saved and printed before being closed.

Again, the **Macro** command is used to send instructions to *Ovation Pro*, in this case to save the fixed document after printing, and to close the window.

```
! Create record of new letter in Letters database.

Clear Letters:Main;      ! Clear card in Letters d-base
Letters:date = Today;    ! Set date field to today
Letters:name = name;     ! Set name field to local name
Copy Letters:;           ! Make a new record from card
```

```
! Copy blank letter template to Letters database.
Local "docname", "scrcfile", "destfile";
docname  = Letters:ref;  ! Use reference as filename.
scrcfile = DataPath("Letters")+".Documents.Template";
destfile = DataPath("Letters")+".Letters." + docname;
*copy (scrcfile + " " + destfile + " ~v~cf");

! Load and merge the document, then fix the data.
MergeTo  "OvationPro";   ! Specify merge application.
DocLoad  destfile;       ! Load document.
DocMerge docname;        ! Merge document.
DocFix   docname;        ! Fix merged data permanently.

! Stop at this point if document is to be edited.
If ( Query("Do you wish to edit the letter?") ) Abort;

! Otherwise save and print document, then close it.
Macro "{savedocument(\"\")}", docname;
Printer  "LaserJet 6";   ! Select printer.
DocPrint docname;        ! Print it.
Macro "{filescrap(currentfile())}", docname;
```

Chapter 18: Exporting data

18.1 Simple export using drag and drop

Impact allows any single displayed data field to be dragged from the displayed card and dropped into any other application that will accept it, such as *Impression*, *Messenger*, and many others. Some applications (such as *TechWriter* and *EasiWriter*) display a 'ghost caret' which indicates exactly where the text will be inserted when it is dropped.

Note that if you hold down ***Shift*** while dragging, the exported data will also be deleted from the field on the card.

If you have partial text selection enabled (see *21.11 Text selection*), you can select part of a field and export that by dragging. ***Alt***-dragging will export the whole field.

Exporting several fields

When a **field map** is active, dragging any field from the card, or dragging from the card background, will export the data from those fields in the map, regardless of which field you drag from.

Alternatively, using the **select tool** it is possible to select several fields in a defined order (click with ***Adjust*** to add to the selection), and export these by dragging to any required destination. This overrides any chosen field map.

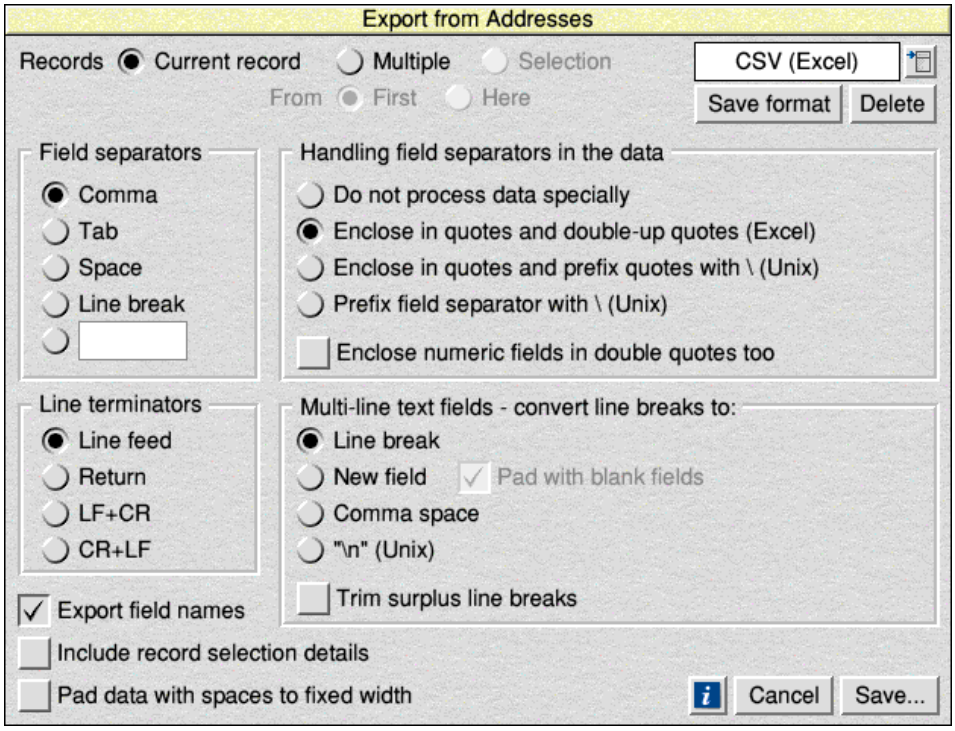
Saving to the Filer

It is also possible to drag a field into a Filer window, to create a text file containing the field's textual content, or to export a copy of an image field. You will be warned if an existing file is about to be overwritten. File names are given as follows:

Selection of fields:	*Selection*
External image field:	the name of the file
Internal image field:	the name of the field or *Image* if the field has no name, with the record number appended.
Other single field:	name of that field, or *Unnamed* if the field has no name.

18.2 Using the export tool

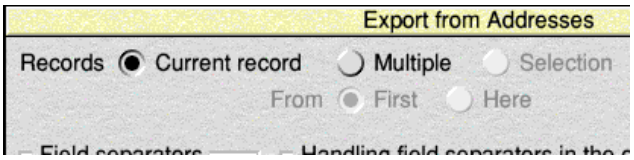
The rest of this chapter describes how to use the **export tool** to export data in plain text, CSV or TSV format from an Impact database. It is opened by selecting the **export tool** from the database toolbar or direct from the menu over a table.



CSV, or *comma-separated value*, and TSV, *tab-separated value*, are standards of data formatting common to many computer platforms, and permit data to be transferred to many other applications, including word-processors, spreadsheets and other database systems. Unfortunately there are many variants of CSV file, and it is important to pick the best one for your target application. See the following sections for further advice on this.

18.3 Selecting fields and records

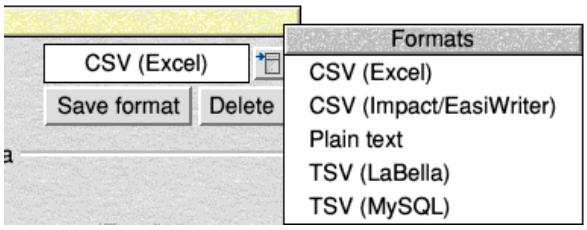
As with the other tools, exporting is done under the influence of any index, filter or field map active in the database. It is therefore possible to export any set of fields in any order and to specify fields to be concatenated into a single field. Any set of records, in sorted or un-sorted order can be chosen.



As first displayed, the *Current record* option is selected. In this mode only the record currently displayed in the card will be exported. If you select *Multiple*, you can choose to export all records (*From first*) or all records starting from the currently displayed record (*From here*). Or you can export a selection of records by opening a **table**, selecting those you want and clicking on the *Selection* option.

18.4 Choosing an export format

The export tool comes with a number of pre-defined export formats so that you can quickly choose the settings appropriate to the task. The export formats can be chosen via the menu button at the top-right of the window.



You can add extra formats to the menu yourself by choosing the settings you want, entering a name in the box, and clicking on the *Save format* button. The formats which are supplied with Impact are protected from alteration and deletion.

By default the first format on the menu will be activated when you first open the export tool. You can choose a different format to be used by default from Impact's Choices dialogue. See *21.10 Default export format* for details.

CSV (Excel)

Export to Microsoft Excel is catered for with a variant of CSV file. The settings ensure that multi-line text files are imported into Excel correctly. If the file is saved with a filename ending '/csv' then when transferred to a Windows PC with Excel the user can usually load the file simply by double-clicking. If you examine the content of the file on RISC OS you will see that multi-line text fields are split across several lines of the file, which looks very odd if you are expecting each line of the file to represent a new record. Excel seems to like it this way, but you won't find many RISC OS applications interpreting this style of file very well.

CSV (Impact/EasiWriter)

This style of CSV file is perfect for export to *TechWriter* and *EasiWriter*, and will result in a table being inserted into the document. It is also a useful format for re-importing to Impact, for example if you are changing the data type of a field or rearranging the order of menu fields (see *Chapter 24: Advanced Impact use*). In order to ensure records are re-imported exactly as they were, the option to *Trim surplus line breaks* is turned off, but can be turned back on if preferred.

Plain text

A straightforward plain text format has been provided, where each new field appears on a new line. Multi-line text fields are exported on a single line, with a comma and space separating the content of each line of the original field of data.

TSV (LaBella)

LaBella is a freeware label printing application which offers a wider range of formatting than *LabPrint*, but which is not so closely integrated with Impact. The best way to transfer data to *LaBella* is using a variant of tab-separated value format.

Please note that *LaBella* (version 6.33) determines the number of fields per record by examining the first line of its input file. If any subsequent lines have more fields, the extras are discarded. The supplied export format therefore has the *Pad with blank fields* option ticked to ensure that all exported records have the same number of lines. However, if you are exporting records with a multi-line address field and a separate postcode field, this then has the effect in *LaBella* of putting blank lines before the postcode on the labels with shorter addresses. Turning off the *Show blank lines* setting in *LaBella* will correct for this.

Note also that by default *LaBella* will strip double quotes that surround fields before display. The export format suggested for *LaBella* does not enclose the data fields in quotes, but if any of the actual fields are surrounded by quotes *LaBella* will strip these too, unless the *Show quotes*

option in *LaBella* is ticked.

TSV (MySQL)

The final export format provided by default is for exporting databases to the open source database server MySQL. This database server is popular on the internet and may be available with your web space hosting. The settings chosen are such that the LOAD DATA INFILE command in MySQL can be used with its default options, without having to specify different field or line terminators.

18.5 Exporting the data

Clicking the *Save...* button will display a standard *Save as* box, from which the file icon may be dragged to any acceptable destination. This may be directly to another application, or into a filer window, etc.

18.6 Formatting the data

Aside from the various export formats supplied with Impact, you have full control over the options used for export, and can save any combination of settings which you need frequently so that they appear on the menu of formats.

Field separators

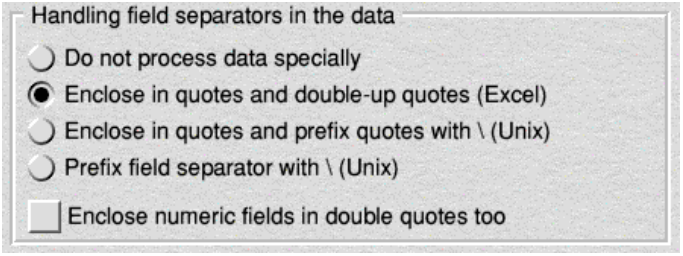
This setting determines what characters are inserted between each field of a record when it is exported. The options are pretty self-explanatory. There are standard options in the form of comma, tab and space. The line break option starts a new line with each new field, and inserts an extra blank line between records. How line breaks are signified in the file is determined by the *Line terminators* option. There is also the option of entering your own custom field separator string. The box accepts codes such as \n for a line feed character, just like entering strings in action scripts and calculations (see *16.4 Control codes in text strings*).

Line terminators

The four options for line termination represent the most common ways of signifying a new line in text files. Most modern systems will accept a plain line feed on its own, but some Windows software requires carriage return plus line feed (CR+LF) and this format is generally required by older MS-DOS and CP/M software. Prior to Mac OS X the Apple Macintosh used line endings consisting just of the carriage return character.

Handling field separators in the data

A problem which affects CSV output in particular is what to do if any of the fields of data contains a comma. There are several approaches to handling this.


Handling field separators in the data
Do not process data specially
Enclose in quotes and double-up quotes (Excel)
Enclose in quotes and prefix quotes with \ (Unix)
Prefix field separator with \ (Unix)
Enclose numeric fields in double quotes too

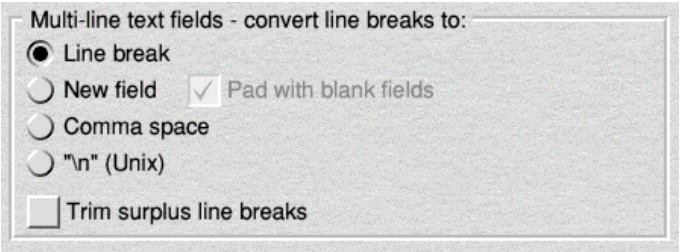
The most common method is to enclose text fields in double-quotes, so that commas within the data cannot be mistaken for the start of a new field, but then the problem arises of what to do if any double-quotes appear in the data. Impact offers ‘doubling-up’ the double-quotes which is

the approach favoured by Microsoft Excel, Fireworkz and many other spreadsheets. Some RISC OS applications, Impact and EasiWriter among them, favour the Unix-style solution of 'escaping' the special characters by preceding them with a backslash. This also entails preceding genuine backslashes with an extra backslash.

There is also the option of simply prefixing the comma (or other field separator) with a backslash to signify that it is not to be interpreted as a field separator but as part of the data field. Again backslashes found in the data require prefixing with a further backslash.

Multi-line text fields: handling line breaks

Multi-line text fields, such as fields containing postal addresses, also present a problem.



The line breaks in the data can be output as they are (this is useful for Microsoft Excel). You can choose to treat each new line of a text field as a separate field in its own right. It will appear in the output with your chosen field separation just like other fields, enclosed in double-quotes as desired. If you are splitting the multi-line text fields into separate fields, you will probably wish to pad the shorter records out with blank fields so that the columns of data continue to align, but this option may be turned off if it suits you to.

The *Comma space* option has some special features. This is designed with plain text output in mind, where the need is for the output to be human friendly, rather than formatted for computers. The multiple lines are joined with a comma and space separating each one, but any commas or spaces included at the end of the lines within the data are stripped off to avoid unsightly double commas appearing.

Most Unix applications, and some RISC OS applications such as Impact and EasiWriter, prefer the line breaks to be shown as a backslash followed by 'n' if they are to be interpreted as line breaks when the file is read back in.

Finally there is the option to trim surplus line breaks from the exported records. If active, this setting causes blank lines at the starts, ends, and in the middle of fields to be removed. Any padding with blank fields (if the *New field* option is in force) is performed after the trimming has taken place.

Other options

You may wish to *Export field names* to provide headings for the data when it is loaded into other software.

The option to *Include record selection details* allows you to keep a note of the field maps, filters and indexes which were in force when the export was performed. It is not really suitable fo use with CSV or TSV output, but is handy for a textual report.

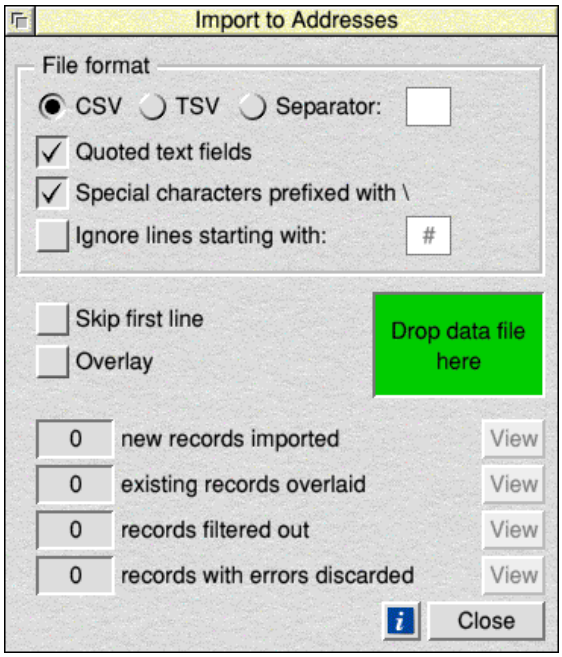
Padding data with spaces to create fixed-width output is handy for display in a text editor with fixed-width fonts, or for use with some really old software applications which do not support CSV. The widths of the output fields are calculated to be just wide enough to accommodate the longest data in the output, so it will vary according to the records you are exporting and will not necessarily be the same spacing if you export a different selection.

Chapter 19: Importing data

This chapter describes how to use the **import tool** to import data into Impact from another application.

Data may be accepted in CSV (*comma-separated value*) or TSV (*tab-separated value*) format. Almost all databases and spreadsheet applications can export data in at least one of these formats. There is also the option to import data with fields separated by a custom character of your own choice.

The **Import window** shown below is displayed when the **import tool** is selected from a database toolbar.



It is easiest to explain the import process by going through a few examples. This chapter therefore starts off by listing the key options briefly. After explaining by example the steps required to import data, there is a further reference section which goes into more depth about the options. Read this after you have looked over the examples, and refer to it as necessary when importing data into your own databases.

Before importing into an existing database we strongly recommend you backup your current data (see *Chapter 22: Making backups*), in case the incoming data is not interpreted in the way you expect.

You can import data directly from another application, if you know what format it is going to be exported in. Before dropping the file onto the *Drop data file here* icon in the Impact import window, make sure the field map and import options have been set appropriately.

19.1 Import options

Tab or comma separators

The normal requirement will be to import CSV data, and the window will be pre-set for this. You may need to use TSV data transfer if the exporting application does not support CSV. If you are unsure whether a file has CSV or TSV data in it, open it in a text editor, such as *Edit*, *Zap* or *StrongEd*, to check whether the fields are separated by commas or tab characters.

In the event of the field separator being neither of these, for example ‘|’ or another character, select the *Separator* option and enter the character in the box provided.

Quoted text fields

Standard CSV files have double quotes placed around text fields, so that commas within the text are not interpreted as field separators. These quotes are normally stripped from the incoming data. If this action is not required this option should be switched off, but it is usually all right to leave it turned on.

Special characters

In CSV or TSV files originating from other RISC OS software, or from Unix systems, there is a convention that certain characters are prefixed with a backslash to negate any special function. For example, a backslash preceding a comma would mean that the comma was part of the field, rather than forming the field delimiter of a CSV file. Again, it is often safe to leave this option active, whatever the source of your file, but it can be turned off if necessary.

For more details about the methods of specifying special characters, see *16.4 Control codes in text strings*.

Comment lines

Rarely CSV files contain comment lines: to ensure these are not interpreted as data you can tick *Ignore lines starting with* and enter the comment character in the box.

Skip first line

Some applications export the data field names as the first line of the file, so this option has been provided to enable that line to be ignored. Impact does not have any automatic mapping from field names contained in your import file to the fields found in your database.

Overlay

If you wish to match existing records in the database with the ones you are importing, and update the fields with the imported version, you need to use the *Overlay* option. Details of this feature are discussed later in this chapter.

19.2 Other settings affecting import

Field maps

Impact needs to know which pieces of data from the imported file need to be assigned to which fields of the database. To do this you need to design an appropriate **field map** and select it before importing the file.

As you will usually require a field map to be active when using the Import tool, a warning will be given if you open the tool window without having selected a field map. If you have no field map active, the incoming fields will be applied according to the internal ordering of the fields in the database.

See the examples below for an explanation of how to construct a field map for importing data, and the reference section for further details.

Filters

If you have a **filter** active when importing data then each incoming record will be tested against the active filter. If it does not pass the filter test the incoming record will be discarded and will not be imported. The filter also has an effect on the operation of **overlay**.

If you want all incoming data to be inserted into the database, ensure that no filters are active.

Indexes

Indexes only have an effect on the operation of **overlay**. During import all local indexes will, of course, be updated to include the new data where appropriate.

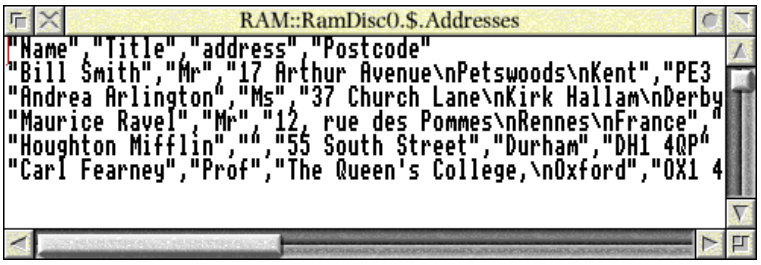
19.3 Example: importing into an existing database

You can follow this example on your own computer by using the files found in the Tutorial directory supplied with Impact 3.40 onwards. These files are also available for download from the Impact web site. Please open the subdirectory named Import to see the example files used in this chapter.

Examine the input file

Usually the first thing to do when importing a file is to take a quick look at it using a text editor, such as *Edit*, *Zap* or *StrongEd*, to see what is in it. We need to work out whether it has comma or tab separators between fields, and whether it is suitable for loading into the current database.

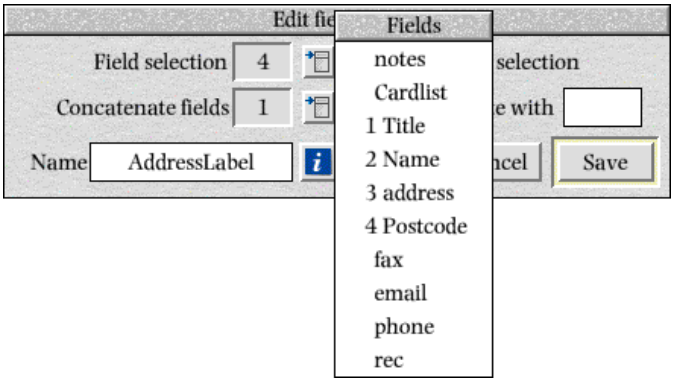
We will use an example file named Addresses. This contains some names and addresses which we might want to add to the sample Addresses database supplied with Impact. You can open the file with your favourite text editor to see what it looks like:



It is obviously a comma-separated value file (CSV): we can see the commas. We can also clearly see that it contains data for the name, title, address and postcode fields.

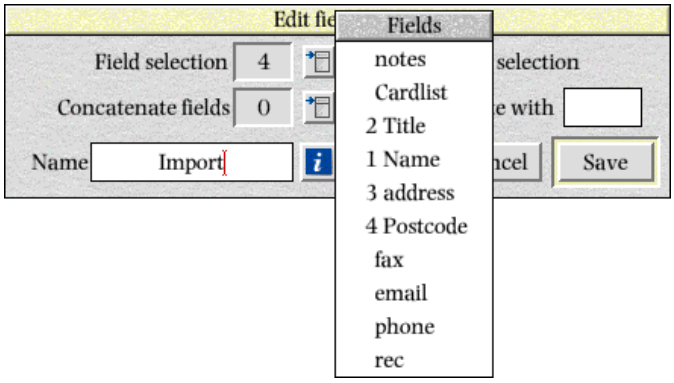
Create a field map

The first step is to create a field map which contains those fields in the right order. We open the Addresses database, and click ***Menu*** over the field map tool. A single field map, AddressLabel, is shown. If we select it, then click ***Adjust*** on the field map tool to edit it, we can see from the field selection menu that although it contains the right fields they are not in the right order for the incoming data:



So we click ***Select*** on the field map tool to deselect the field map, and then click ***Adjust*** to create a new field map. We select the four fields we want in the correct order on the field selection menu (clicking with ***Adjust*** to select a field keeps the menu open to allow us to select the next). Concatentation of fields is ignored for import purposes, so all we have to do is enter a name for the field map, such as 'Import' and click *Save*. Here is what the field selection menu

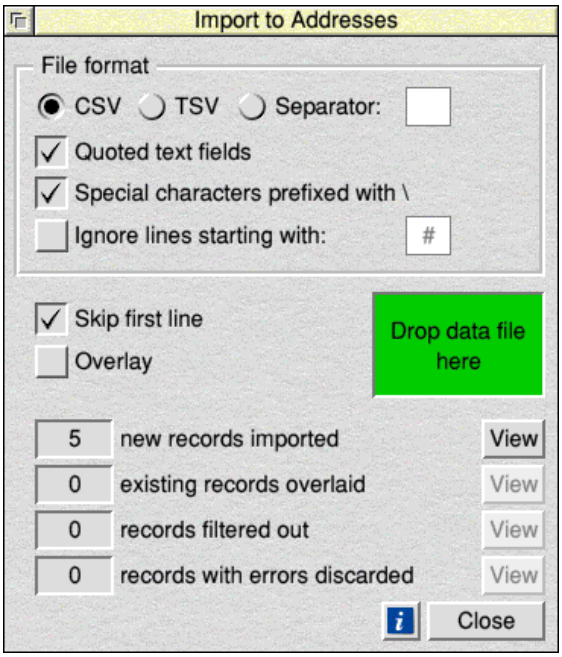
looks like for the new field map when we have finished:



Import the data

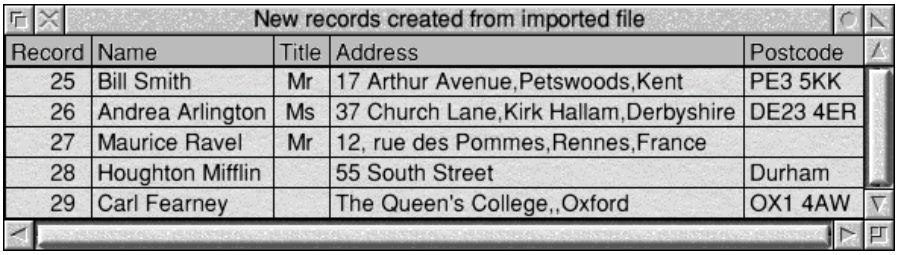
Make sure the 'Import' field map is selected and open the Import tool. We need to set up the tool for CSV import, with quoted text fields. We need to skip the first line in the file because that was indicating the field names: we do not want to create a new record with a name of 'Name' and an address of 'address'!

Drag the Addresses CSV file and drop it in the area indicated to import the records. The import tool dialogue is updated to show that five records have been imported:



Check the results

Click on the *View* button to show the new records in a table.



Record	Name	Title	Address	Postcode
25	Bill Smith	Mr	17 Arthur Avenue,Petswoods,Kent	PE3 5KK
26	Andrea Arlington	Ms	37 Church Lane,Kirk Hallam,Derbyshire	DE23 4ER
27	Maurice Ravel	Mr	12, rue des Pommes,Rennes,France	
28	Houghton Mifflin		55 South Street	Durham
29	Carl Fearney		The Queen's College,,Oxford	OX1 4AW

If you compare it to the original CSV file you will note some differences. The Title field in the

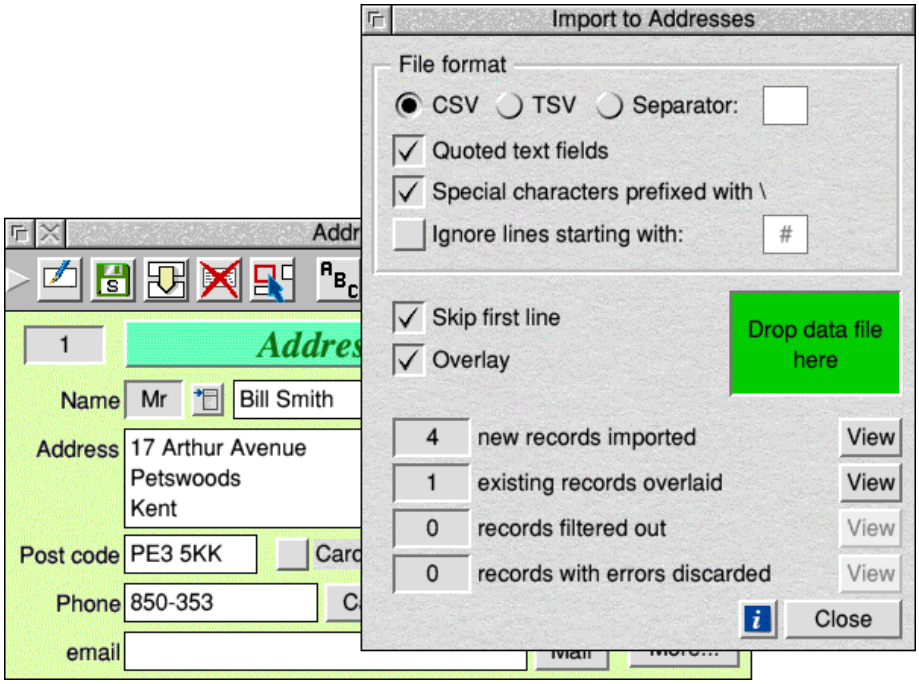
Addresses database is a menu field, and has a limited range of options, which did not include 'Prof'. Professor Carl Fearney has therefore been imported with no title. You might want to rectify this by editing the database design and adding 'Prof' as an option for the title menu field.

The entry for Houghton Mifflin has the town in the postcode field, and the postcode has not been imported. On inspecting the CSV file we see that the imported line is defective: the address has not been entered in one field but is split into two, meaning that there actually five distinct fields listed for that line. Impact will silently discard any fields which are surplus to the requirements of the field map.

Another problem is that we now have two Bill Smiths in the database, as the first record in the database was also a Mr Bill Smith of Petswood. It is clear that he has changed his address, and for this we should have used the Overlay feature to match and update existing records.

Make corrections and import again

If we are unhappy with the import, we might want to make adjustments and try again. The quickest way to delete the new records is to choose *Select all* from the menu over the table of new records and then *Delete selection* from the menu. We can now tick the overlay option and import the file again. This time the dialogue box shows four new records imported and one overlaid. Sure enough the address and postcode of record 1 in the database have been updated to reflect Bill Smith's new address:



Note that when matching records with the overlay option, only the first field in each row is considered. In this example the name of the individual must be exactly equal for the record to be overlaid with new data. Any fields not included in the field map, such as the telephone number, are unaffected by the overlay process.

Techniques for overlaying data are discussed in further detail below, and we recommend that you read this part of the manual in full if you need to import data with overlay.

19.4 Example: importing a file to a new database

You may want to create an entirely new database using a file exported from another application. You will first need to design and create a new database to receive the imported data. This example takes you through the various decisions you would need to make in such a situation.

You can follow this example on your own computer using the Tutorial directory supplied with Impact 3.40 onwards. These files are also available for download from the Impact web site. Open the Import subdirectory and locate the file named Members. This contains some data that might form part of a membership database for a society.

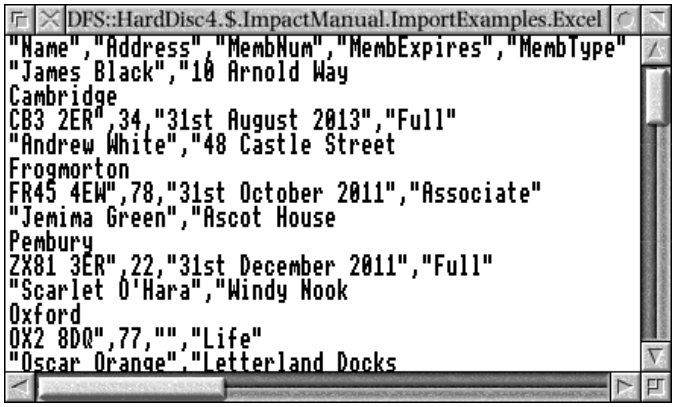
Examine the input file

Firstly we take a look at the file to be imported using a text editor:

ADFS::HardDisc4.$.ImpactManual.ImportExamples.Tutorial.Import.Members
"Name","Address","MembNum","MembExpires","MembType"
"James Black","10 Arnold Way\nCambridge\nCB3 2ER",34,"31st August 2013","Full"
"Andrew White","48 Castle Street\nFrogmorton\nFR45 4EW",78,"31st October 2011","As
"Jemima Green","Ascot House\nPembury\nZX81 3ER",22,"31st December 2011","Full"
"Scarlet O'Hara","Windy Nook\nOxford\nOX2 8DQ",77,"","Life"
"Oscar Orange","Letterland Docks\nLetterland",102,"30th November 2011","Junior"
"Robber Red","107 Robber Towers\nThieftown",1,"31st March 2009","Junior"
"Duke of Burgundy","Chateau de Moutard\nDijon",214,"31st December 2011","Full"

We see that the fields are separated by commas, and the first line contains what appear to be the names of the fields in the system which has produced the file. A line listing field names is not compulsory in a CSV file, and if it were not present we would have to look at the lines of data to try to work out what the fields might be.

There is a single address field, and in the records we can see that line breaks are indicated with '\n'. This is a convention more commonly found with RISC OS or Unix software. If the CSV file had come from a Microsoft Windows system then any multi-line data would be exported with actual line breaks, meaning that the records might be split over several lines of the file. Here is how that would look in the text editor:



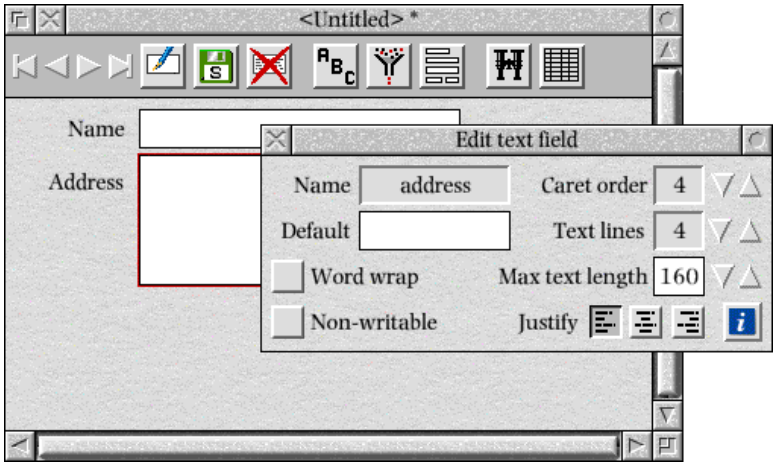
Fortunately Impact correctly handles each type of file without any user intervention. The important point to note, though, is that we will need to create a multi-line field to hold the address. This is good news, as on the whole it is easier to handle address data in Impact using a multi-line text field than in a series of separate fields. If you subsequently need to add capacity for a longer address, it is easy to edit the card to add space for another line, but if you store the lines of the address each as separate fields, adding an extra field is quite an undertaking as the new field might need adding to field maps and merge documents too.

We would need to scan through the file to get an idea as to how long the text fields will need to be. When creating a text field, the upper limit on the size of the field must be set. It can be adjusted easily later, but if it is set too small then the field values will be truncated on import. Impact will warn you during the import if any of the values have had to be shortened, and by how much, so it is easy to delete the imported records, adjust the field size, and import again if necessary.

Create a new database

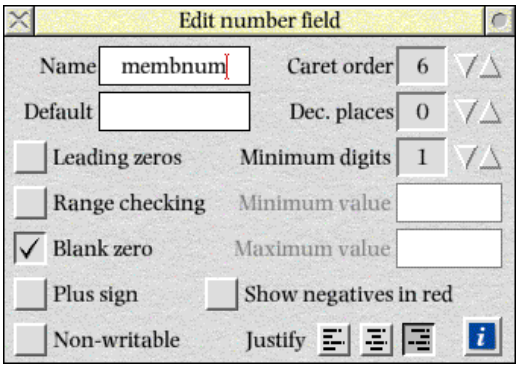
At this point we might start creating the database to receive the imported data. You can work through this example if you wish, or you can just study the example provided by dragging the directory 'MembersDB' to Impact on the iconbar.

If you want to work through the process yourself, start by using the *Create...* option from the iconbar menu, fill in the database name as you choose, and drag a couple of label and two text fields to the card.



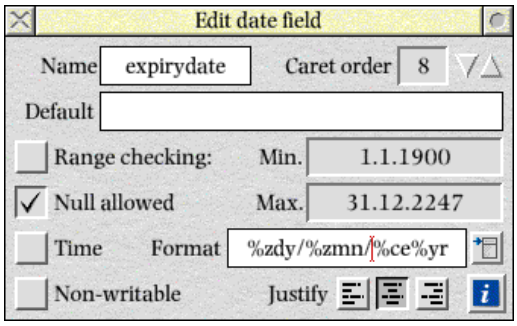
The picture shows how we might set up the name and address fields, with four text lines and a generous 160 character limit for the address field. To examine the field definitions in the 'MembersDB' example, switch to card design mode by choosing *Edit card* from the menu over the card.

Now we return to the CSV file and see what we might do with the other fields. The next is called 'MembNum' and it appears that all the records have a number in this position. We could use a number field, or perhaps a sequence field here. There is nothing stopping us using a text field, however, if we think we might want to have text in the field in the future. The following picture shows how we might set up a suitable number field:



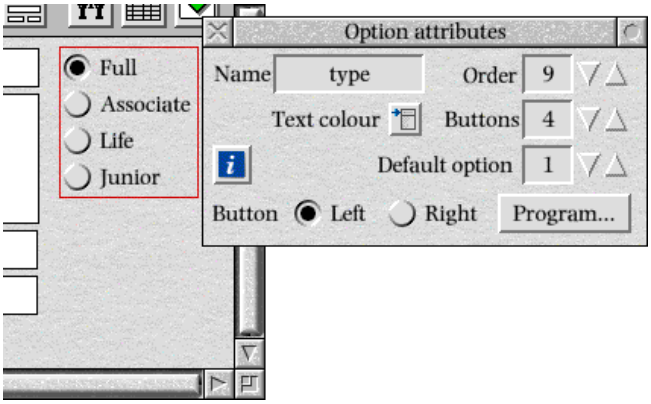
Note that the number of decimal places shown has been set to zero, and the 'Blank zero' option is ticked, which will leave the box empty if nothing (zero) is entered in the card.

The 'MembExpires' field clearly needs to be set up as a date field if we are going to do anything useful with it. Impact is fairly tolerant about date entry formats, but one thing which is insisted upon is that the different elements of a date are entered in the same order as the display format chosen for the field. In the example CSV file they are in the order date, month, year, so providing we choose that order for the display of the date field in our new database we should be all right. The import file has the month in words, but Impact will still read it properly even if the display format has the month in digits.



Here I have set up the date to display in the format dd/mm/yyyy.

The 'MembType' field looks like it has a limited range of values: Full, Associate, Life and Junior. We might want to set up a **menu** or **browser** field to receive these, or a set of radio buttons (**option** fields). But we could equally well place these values in a text field. In a more complex example, the membership type might actually be a foreign field, linking to another database where the subscription, voting rights, etc., of that type of membership can be looked up.



In a real life example you might well play with several different database designs before settling on the one which will work best.

Add the import tool

Finally, do not forget to add the Import tool to the database toolbar: it is not there by default. Double-click on the toolbar when you are in database design mode to open the tools window, or use the menu over the card.

Save the new database design

Once you are happy it is time to save the card design and to try importing the file. In the **field selection window** enter a database name, such as MembersDB, and then click *Save*. The database is now ready for use.

Import the data

You can now import the data from the file into the newly-designed database. The process for this is exactly the same as importing into an existing database as shown in the previous example.

If we have created the fields on the card in the same order that they appear in the import file, there is no need to create a field map, and you can ignore the warning when opening the import tool. Tick to skip the first line (containing the field names), and drop the CSV file onto the import tool window. Once the import has completed, you can view the new records in a table

by clicking on the *View* button.

19.5 Further reference

Field maps

It is unlikely that the fields in the data being imported are in the correct order required by the local database. This problem is resolved in Impact by using an appropriate **field map** active on the database. This allows the incoming data to be inserted into specified fields in a defined order.

Because a field map is usually required, Impact will warn you if you open the tool window without having selected one. With no field map active, the incoming data will be assigned according to the internal ordering of the fields in the database.

Any concatenation settings in the field map will be ignored during import: the import process will expect those fields to appear separately in the imported file.

If the field map lists more fields than actually appear for any record in the file, then the data will be read for as many fields as possible, and the remaining fields will be imported as though the file contained a blank for those fields. Number fields would thus be set to zero, and a date field to a null value.

Conversely, if any records in the file being imported have more fields than are present in the field map, any data from the surplus fields will be discarded silently.

If the field map contains calculation fields then data for these will be expected in the file being imported, but will be skipped over and not written to the records created or updated during the import process. The same is true of foreign dependent fields. Either of these field types may be used as the first field in the field map for the purpose of record matching when overlaying data, however.

Overlay

When importing records using overlay, Impact assumes that the first field in each record will uniquely identify it, and so will search the existing database for a record having the same content in this field. If one is found, that record will be over-written with the incoming data. Fields not included in the import will not be changed. If no matching record is found, a new one will be created, as with a normal import.

When you are attempting to overlay data from a source you have not used before, it is wise to test it by importing with overlay turned off initially, so that none of your existing records are altered. If you happen to have got your field map wrong and data are ending up in the wrong fields, you can easily select the new records and delete them again, but any changes to existing records are harder to repair.

Overlay should be used with great care, as the first field may not be unique in your existing database, so you may find the wrong records are replaced. For example, if the first field is a person's surname, in all but the very smallest databases there are likely to be duplicates. Beware also of matching on a field whose value might conceivably change: a person's surname being a case in point.

If the field you want to match on is not the first in the file you are importing, you may be able to rearrange the file, prior to import, by using a tool such as Ray Favre's *CSVamp* or a spreadsheet. You could even import into Impact temporarily, without overlay, and re-export the data with the fields in the desired order.

A value which is guaranteed to match uniquely is the Impact record number, and this will not change for the life of the record. A sequence field is almost as good. But these are only likely

to be available if you are importing data from a source which obtained the records from your Impact database in the first place. To match by record number you would need to create a calculation field of numeric type, with the calculation expression being "Record". See the Videos and Addresses sample databases for examples you can copy.

If you do not have any field which is sufficiently unique to use for matching, it is possible, but fiddly, to combine two or more fields for matching. For example, you might have a surname field and a forename field, and the two together might be sufficient to distinguish all your records. First of all, import the data as new records, without overlay being turned on. Next, create a calculation field, whose expression is along the lines of:

```
surname + ":" + forename
```

Create a new field map containing all the fields you had imported from the file, but with the new calculation field first in the field map. Using this field map, export the new records to a CSV file, and then delete all the new records from the Impact database. At this stage you will have your original data, unaltered apart from the extra calculation field, and a CSV file whose first field uniquely identifies records for matching! Using the newly defined field map, import this CSV file with the overlay option turned on.

Filters and overlay

If you have a filter selected when using the import tool, then any incoming records which would not pass the filter are discarded and will have no effect on the records in the database.

But when overlay is in operation, the current filter also affects which records are searched for possible matches. Any record which does not pass the current filter will not be considered for matching, and will not be overlaid. Instead a new record will be created, providing the proposed new record from the import file passes the filter test, of course.

Indexes and overlay

The currently selected index will only have an effect on the import operation if you are using **overlay**. The index will affect the order in which the records are searched for a match, and, in fact, which records are even considered for a match with the incoming data in the first place.

Without an index being selected, the value in the first field of each incoming record will be searched for in the database, starting with the first record in the database and working through until a match is found. If an index is selected, the order will instead be according to the chosen index. Only the first matching record will be modified, thus if the field value matches more than one record currently in the database, the choice of index may determine which record is actually overlaid. If a further record in the imported file has the same first field value, it will match the same record again, not any subsequent record further down the index.

If the current index applies a filter, or is a selection index, then there will be an even more dramatic effect on overlay, because only the records included in the index (by either forming part of the selection, or passing the index's filter test) will be searched for matches. If a record has its first field matching a record in the imported file, but yet is not included in the index, then overlay will not occur and instead a new record will be created.

This aspect of the import process is rather complicated, and for the most part users will get better results by ensuring that there is no index selected when using the import tool.

19.6 Import and field types

Data type translation

Within a CSV or TSV file, all data, including numeric values, is stored as text. During the

import process, each field will be converted into the data type appropriate to the field it is entered into. For example, text fields are normally listed between double quotes in CSV format files, but Impact will attempt to translate these into another form, such as numeric or date, etc., if necessary. Similarly, where numbers are found in the input, these will be treated as text if they need to be entered into a text field.

Note that during this process, if a text field is read into a number field, an attempt will be made to interpret the given text as a number. If the text does not start with digits, a decimal point, or a plus or minus sign, then the outcome of this will be a zero in the number field.

If you are importing date fields, check the format of the dates in the input file, and edit the date fields in your database to use the same format (or at least one which is in the same order), or Impact may not interpret the dates correctly. You can always change the date format for the field back again, after finishing the import job, so that the dates are again displayed in the way you prefer.

Internal file fields

If an internal file field appears in the field map, data will be imported by creating a text file containing the appropriate value from the imported file. If the imported value is blank, then no file will be created, and if an internal file existed for an overlaid record it would be deleted.

It is not possible to have new internal files created with a filetype other than text via the import process: you would have to alter them manually afterwards if necessary.

Browser and menu fields

The values expected for import into browser or menu fields are the text strings which appear in the field or menu display. The imported value is matched case-insensitively against each possible value for the browser or menu field, and the first one which matches is used for the imported record. The whole of the value must match: truncated values do not match. If the imported value matches none of the possible values of the browser or menu field, then the field will be assigned the first value in the list, not the default value.

Flag and option fields

As for browser and menu fields, the values expected for import are the text strings which appear by the tick boxes or radio buttons on the record card. Matching is case-insensitive, and the whole of the string must match the whole of the button labels.

In the case of option fields, the first option which matches is selected, and if no options match, none of them will be selected on the resulting record.

Flag fields are more complex. Here several flag values belonging to the same field may be given, separated by commas. Any which are matched by labels on the card will have their boxes ticked. in the resulting record. Obviously the commas separating the flag values in the import file must be distinguished from the comma used for field separators, either by having the whole set enclosed in double quotes, or by the commas being escaped by a preceding backslash.

Importing into foreign fields

It is possible to import data into foreign key fields. This will *not* import data into the foreign database, but will search the foreign database for a matching record, and establish the relational link.

For this to work, there must be a matching record in the foreign database, so if you are going to import a file into the foreign database too, you should do that first. If no matching record is found, then no link is established, but no error message is given.

If foreign dependent fields are included in the field map, then data for these must appear in the file, but will be skipped over and not written to the records created or updated during the import process.

19.7 Viewing the results of an import

The Import window includes a *View* button beside each reported count of records. For records which have been imported or overlaid, clicking the *View* button displays the records in an Impact table. This enables you to check that the data have been imported correctly, and if there are any problems the imported records may easily be deleted again. Note, however, that deleting a record which has been overlaid will not return you to the previous version of the record, which is why it is advisable to take a backup of your database before overlaying.

For records which have been discarded, either because they failed a filter test, or because errors occurred, clicking the *View* button will open a file listing them in your text editor so that you can examine and amend faulty records where appropriate. This file is only temporary and will get overwritten by Impact during the next import, so if you wish to edit these records and re-import them, you will need to save the file to a new location first.

Chapter 20: Caller display

This chapter describes the support given to the caller display unit and software available from Octopus Systems.

20.1 Background

When a phone call is about to be received, the phone company send a coded message down the phone line giving the phone number of the subscriber who is calling. This may be received and used in various ways.

This service may be available for free or be subject to a charge from your phone company.

20.2 The Caller display unit

This is a small box having a standard phone lead on one end, and a serial port cable on the other. In use, one end is plugged into the phone line, usually via a two way splitter, and the other end into the serial port on the computer.

When a call is detected, the data is converted to ASCII characters and sent to the computer via the serial port.

20.3 Serial ports

If your serial port is already being used by a modem, for example, it will be necessary to install extra serial ports by purchasing a serial port card, many of which are available on the general market.

20.4 Support software

The software supplied by Octopus Systems is fully described in its own manual, and is not covered here.

One of the functions of this software is to broadcast a standard Wimp message to all running applications on the computer, containing details of the call being received. This message is recognised by Impact, which may be configured to take certain actions.

20.5 Looking up the number

When the message is received, Impact will open and load a named database, and attempt to find the supplied phone number in a named field. The database and field name must be supplied from the **Choices** window available from the iconbar, as described in Chapter 21: *Choices*.

If a match is found, the record card will be displayed on the screen.

20.6 Taking action

Impact will also execute an **action script**, as defined in the **Choices** window. This must specify a database and action script name.

The action button may, for example, insert the time/date and phone number in a database of logged calls, enter the time and date into a customer contact file, etc.

For example, assuming the call log database is called *CallLog* and the number was looked up in *Addresses*:

```
Clear CallLog:;
CallLog:date     = Today();
```

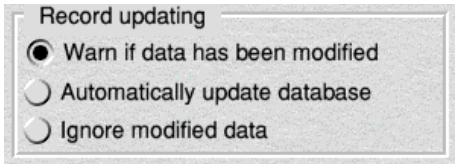
```
CallLog:number   = Phone();
CallLog:name     = Addresses:name;
Copy CallLog:;
Addresses:called = Today();
Save Addresses:;
```

Chapter 21: Choices

This chapter describes the user definable choices available which allow you to customize Impact to suit your own preferences.

The **Choices window** is displayed by selecting *Choices...* from Impact's iconbar menu. This is a scrollable window with sections referring to various parts of the program.

21.1 Record updating



The first three buttons in this window control what happens when browsing or selecting new records, when the displayed card data has been modified. The effect of each button is described below.

Warn if data has been modified

If this is selected a warning message will be displayed whenever a modified record card is about to be replaced by another, or the database closed.

Clicking on *Cancel* in this message window will cause the current action to be cancelled so that the modified data may be saved, before continuing.

Clicking on *OK* will cause the modifications to be lost, and the new record to be displayed.

Automatically update database

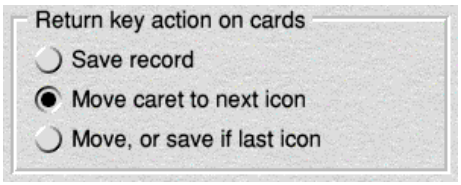
If this option is selected, no warning will be given, but the modified data will be automatically saved before the new record is displayed.

Ignore modified data

With this option selected, the modified data will be lost, and the new record displayed. No warning will be given. In this mode any changes must be explicitly saved before attempting to select a new record.

21.2 Return key action on cards

There are three options available for the behaviour of the ***Return*** key when you are editing data on record cards.

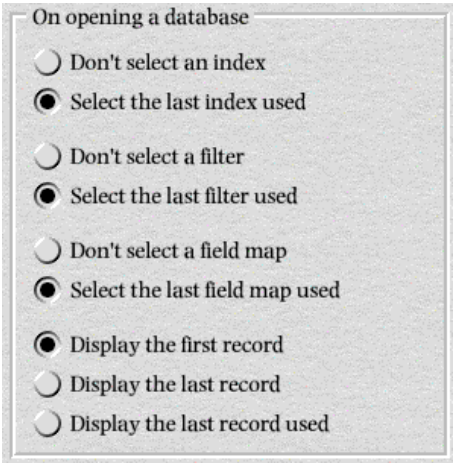


If you choose *Save record* then the current record will be saved whenever you press the ***Return*** key in a writable icon on the card. You can instead choose to have the caret move to the next icon when ***Return*** is pressed. Finally there is the option to have the caret move to the next icon until the final icon is reached whereupon the record will be saved.

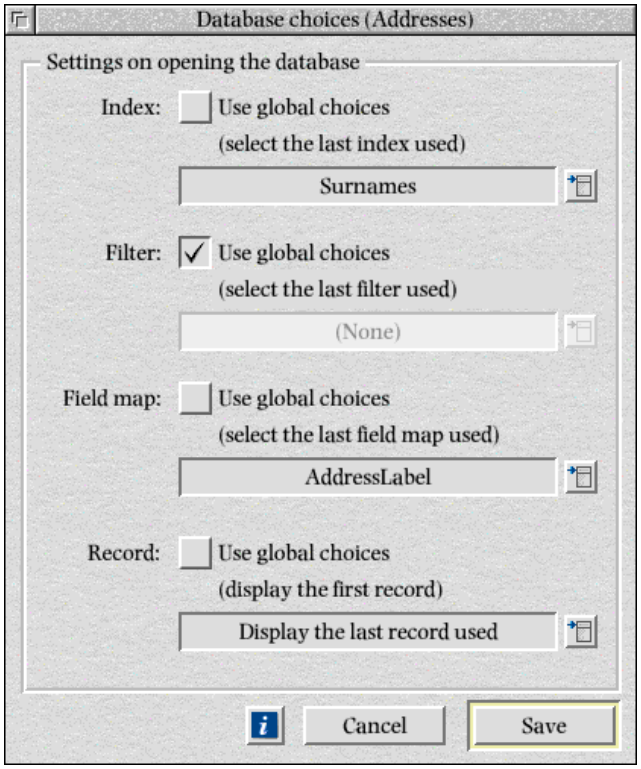
Whichever option is chosen, pressing ***Return*** when in a multi-line text field will insert a blank line and move the caret down to the start of it. Saving or moving to the next writable field will only take effect when the caret has reached the last line of the text field's box.

21.3 On opening a database

There are four settings which you may wish to have control over each time a database is opened: the choice of index, filter, field map and record, to avoid having to select these every time you want to use your databases.


On opening a database
Don't select an index
Select the last index used
Don't select a filter
Select the last filter used
Don't select a field map
Select the last field map used
Display the first record
Display the last record
Display the last record used

These are set at a global level from the main choices window (above) and may be overriden locally for each database by choosing *Database choices...* from the **card menu**, and unticking *Use global choices* for each item as required in the dialogue box (below).



For indexes, filters and field maps the global choices option gives the choice between selecting no index/filter/field map or selecting the one which was in action when the database was closed (which may be none). In the local choices for each database you can also choose from the list of indexes, filters and field maps a particular one to be in force when the database is opened.

For records the global and local choices options both allow you to select the first or last record (according to the index and filter in use) or the record which was in use when the database was closed.

21.4 Backup

The action of the backup facility, described in *Chapter 22: Making backups*, may be controlled from this section. The first three buttons control what happens when a database is opened or closed which has been modified since its last backup. Note that for each database backups must first be enabled using the **backup tool** as described in the next chapter before these choices will have any effect.

Warn

With this selected, a warning message will be displayed informing you that the database has been modified since its last backup.

Auto

Selecting this option will cause the database to be backed-up automatically if required whenever it is closed.

As only a limited number of backups are stored, it is advisable to use this feature in conjunction with the *Timed* option detailed below. This will mean that your saved backups cover changes in your data over a reasonably long period of time, thus enabling you to restore an old version of the database several days after a drastic alteration was made. You can, of course, make a backup manually at any point if you wish, using the database's **backup tool**.

Ignore

If you select this option, the program will not warn you, or do a backup automatically. It will be up to you to create backups whenever you like.

Backups saved

This sets the number of past backups which will be stored. When this count is exceeded, the earliest backup will be overwritten with the new.

Timed

Selecting this option encourages backups to be made at set intervals chosen from *half daily*, *daily*, *weekly* or *4-weekly* using the adjuster arrows. If a database has been modified since its last backup then you will be warned (if *Warn* is selected), or a backup made automatically (if *Auto* is selected), only if the set time period has elapsed since the last backup was made.

21.5 Field menus

These options define the sort order used for field names on menus throughout the program.

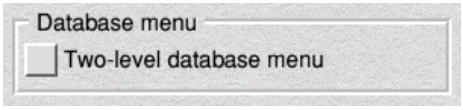
Sort by name

Selecting this causes all field menu items to be displayed in alphabetical order.

Sort by order

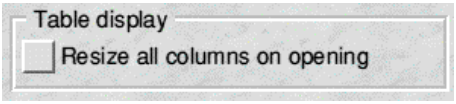
Selecting this causes all field menu items to be presented in their **caret order**, as set during card design in the field's attribute box.

21.6 Database menu



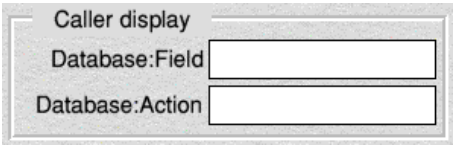
By default all databases known to Impact will appear on one long menu. They are grouped according to the directories in which they are held. You will usually have a *UserData* directory and a *SampleData* directory, the latter containing the demonstration databases supplied with Impact, but it is possible to set up further directories which Impact will examine to find databases to list on the menu. If you have a large number of these, a two-level database menu can be beneficial. The top level menu will then list the directories such as *UserData* and *SampleData*, and the databases contained in each will be listed on sub-menus.

21.7 Table display



Ticking *Resize all columns on opening* will ensure that all tables will have columns wide enough to display the whole of their data. Calculating the column widths may be slow for a large database on old hardware, so this option is not selected by default.

21.8 Caller display



This feature controls the program actions when a telephone call is received, if Caller display is in use (see *Chapter 20: Caller display*). The required syntax is as shown in the item label, and is described in *Chapter 16: Calculations*.

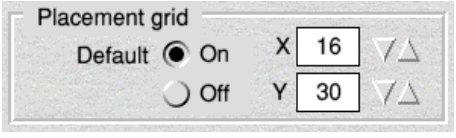
Database:Field

This specifies the database and field which will be searched for a matching phone number.

Database:Action

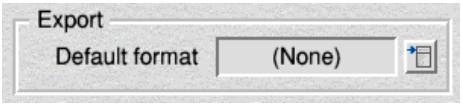
This specifies the database and action field which will be executed when a call is received.

21.9 Placement grid



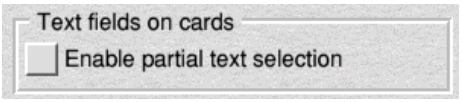
These settings control the *x* and *y* spacing of the placement grid used for card design editing, and whether it is on or off by default.

21.10 Default export format



You can pick which of the export formats will be selected when the export tool is first opened. If you do not make a choice then the first option on the formats menu will be chosen. See *Chapter 18: Exporting data* for more information.

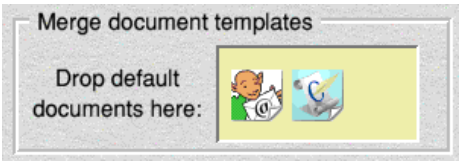
21.11 Text selection



You may wish to be able to drag to select part of the text in your text fields, for copying & pasting, dragging & dropping into other applications, etc. This is not enabled by default. To enable partial text selection, tick the box. Partial text selection only applies to text fields and not to other writable icons.

Enabling text selection affects the action of various mouse clicks and drags: for full details see the table in *Appendix 2: Mouse and key actions in record cards*.

21.12 Merge document templates



In order allow Impact to create a new merge document from the **merge tool**, it needs to have a default document to copy. This section of the Choices window shows which applications have default documents available. A default *ImpEmail* message is provided with Impact, but can be overwritten by dropping a replacement on this part of the Choices window. To add a default document for another application, simply create one and save it to this part of the Choices window.

To open any of the default documents for editing, just double-click on the appropriate icon in the Choices window.

21.13 Implementing changes

To implement any changes made to the settings in this window, click on the *Set* button.

If you wish to save the settings so that they are used the next time you use Impact, then clicking on the *Save* button will both implement any changes and save the settings to disc.

If the *Cancel* button is clicked, the window will be closed and any changes ignored.

Chapter 22: Making backups

This chapter describes the use of the **backup tool** available from the database toolbar, and how to restore a database from a saved backup.

Backups may be stored on a pre-defined filepath, or by dragging a directory icon to any desired destination. Any backup may be restored in a similar manner.

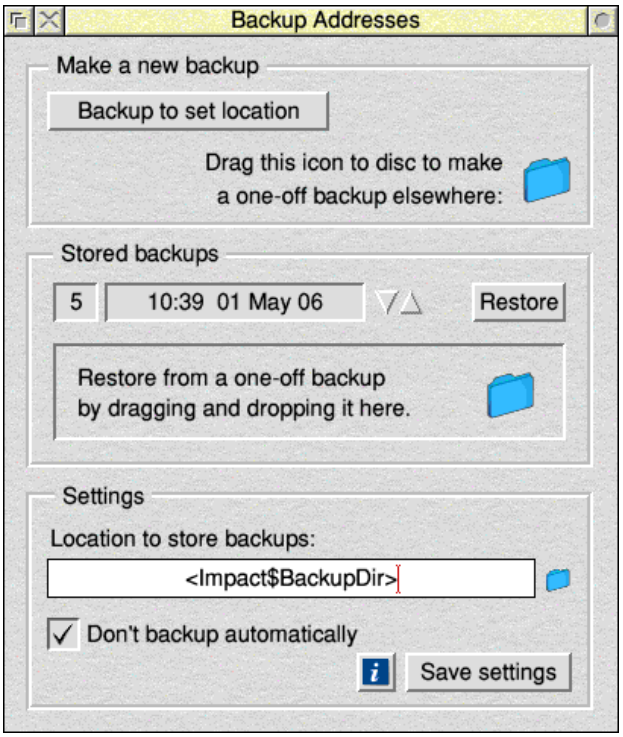
All backup copies are compressed to save disc space, the compression ratio being near to 50%. Compression and de-compression are automatic.

22.1 The need for backups

Making systematic backup copies of all databases is an essential part of all computer management operations, from large institutions to the single home computer. There are many ways in which data may be lost or corrupted, from a simple power failure or computer crash, to computer theft, fire, a disc drive fault, a damaged disc, or other catastrophe.

22.2 The backup window

Selecting the backup tool from a database toolbar will display the **Backup window**.



The *Stored backups* section will show the date of the last backup done, and the *Settings* section will show the backup location for this database (which is likely to be blank if the database has not been backed up before).

You may cycle through the previous backups, which will be presented in date order. The highest number will always be the last one saved, and have a maximum value equal to the number of backups set in the **choices window** (see *Chapter 21: Choices*), as earlier backups are removed. To store more backups you will need to alter the *Backups saved* setting in the choices window.

22.3 Setting the backup location

By default, a newly created database will have the location to store backups set to

```
<Impact$BackupDir>
```

which is a system variable set up in Impact's *!Run* file. It normally points to a directory called *BackupData* in the same directory as the *!Impact* application itself.

You may choose to set the backup location to something else: just find or create a suitable directory on your hard disc. You may use the same location for as many databases as you like, as each will be stored under its own name. You may type the new location name into the box, or drag the directory from the Filer window to the box. Alternatively, open the Filer window showing the contents of the backup directory, and drag the small directory icon from the Backup window to that window.

If you wish to use a floppy disc for your regular backup path, the easiest way to set this up (if you are using RISC OS 4 or above) is to insert the disc into your floppy drive and drag the small directory icon to the floppy disc icon on the iconbar. Note that Impact will not be able to make automatic backups to floppy disc without your inserting the disc in the drive!

22.4 Making a backup

The easiest way to make a backup is to click on the *Backup to set location* button. In this case the backup will be made to a location on the specified path, and an earlier backup removed if the total backup count has been exceeded.

Alternatively, the top directory icon may be dragged directly to any Filer window, e.g. onto a floppy disc. In this case the backup count is ignored and a single backup is performed which will overwrite any backup existing in the same location.

22.5 Automatic backups and warnings

In the **choices window** (see *Chapter 21: Choices*) you may choose to have backups done automatically if needed when a database is closed, by choosing the *Auto* option. This setting will apply to all databases.

Note, however, that each individual database you wish to have backed up automatically will need to have the backup tool installed on its toolbar, the backup location set and the *Don't backup automatically* option unticked in its Backup window.

It is possible to set backups to be performed at weekly or other intervals instead of every single time a change is made, using the *Timed* option in the Choices window (together with the *Auto* option), which may provide a more useful variety of backups to restore from later.

If you prefer to make all your backups manually, you may wish to be warned when your databases require backing up. Choose *Warn* in the Choices window, and ensure that *Don't backup automatically* is not ticked in the database's Backup window. This will mean that if you close a modified database without first making a backup, or open it again, a warning will be given that the database has not been backed up.

If you do not wish to be warned or to have automatic backups for any database, choose *Ignore* in the Choices window.

22.6 Backup all

From the main iconbar menu you can choose the option *Backup all*, which will back up all the databases which are currently open. Databases which are not currently open will not be backed up by this process. Again, each individual database must have the backup location set in the backup window if this operation is to produce a backup.

22.7 Restoring data

Clicking the *Restore* button will restore the backup displayed, which will be latest one when the window is first displayed. A warning will be given reminding you that any data entered since the backup will be lost, and the restoration will be carried out if you click *OK*.

If required, any previous backup may be selected by scrolling through those available before clicking the *Restore* button. If you do this, a warning will be given asking you to confirm the action. You will also be given the option to remove any backups which are dated after the one being restored.

To restore from a backup not saved on the specified path, simply drag the directory containing the backup to the appropriate area of the backup window. A warning will be given asking you to confirm the action.

22.8 Restoring a database which cannot be opened

If a database has become corrupted through some fault it may not be possible to open it in order to use the **backup tool**. Providing a backup has been made at some point in the past, it is possible to restore from this backup in the following way.

Open the directory where your database is stored (usually called *UserData*, in the same directory as the Impact application) and find the directory which bears the name of your database. Rename it to something else, or move it to a different location on your hard disc.

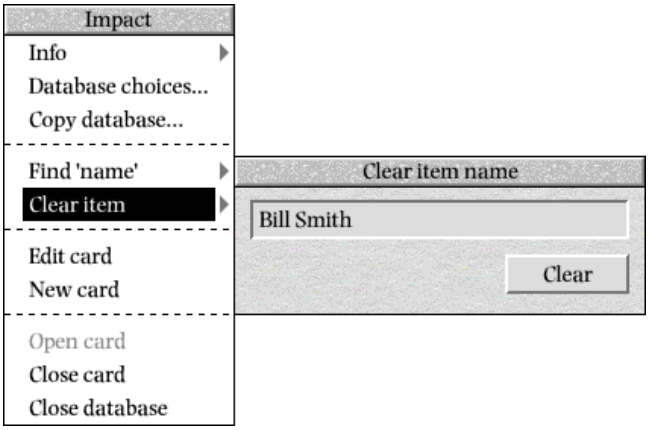
Now in Impact, choose *Create...* from the iconbar menu to create a simple database with the same name as your faulty database (i.e. the name it had before you renamed it). All you need to do is place the backup tool on the toolbar and a single data field on the card (e.g. a text field) and give the database the correct name. Save your card design and click on the backup tool. Once you have set the backup path to the right location you should now be able to restore from a backup of your original database in the normal way, overwriting the card you have just created.

Chapter 23: Deletion

23.1 Clearing field contents

Writable fields are most easily cleared by placing the caret in the icon and pressing ***Ctrl-U***.

File, **image** and **foreign key** fields may have their contents cleared in the current record by following the arrow by *Clear item* from the **card menu**, and clicking on the *Clear* button. Clearing a foreign key field will also clear any dependent fields.



23.2 Deleting records

Individual records may be deleted using the **delete tool** available from the toolbar. Single records or multiple records may also be deleted from a **table display** by selecting them and choosing *Delete selection* from the table menu.

23.3 Removing fields from a card

Before you remove fields from a database card, we strongly recommend making a backup of the database (as described in *Chapter 22: Making backups*) in case anything goes wrong as a result of your deletion.

To remove a field, choose *Edit card* from the card menu to bring up the card editing window. Click ***Menu*** over the field you wish to delete and choose *Delete* from the *Field* sub-menu. You will be warned if this is the only occurrence of this field, so that you can change your mind. Then click on *Save* in the Field Selection Window to save the updated card design.

The data from the deleted field will be removed from the disc. If the field was used in a filter, index or field map, these will be updated or deleted as appropriate.

Impact will not stop you from removing fields which have been used as foreign fields in other databases. Think before you delete!

23.4 Deleting a card

If a database has more than one card defined, you may delete any of these cards by choosing *Edit card* from the card menu to bring up the card editing window and clicking on *Delete card* in the Field Selection Window. You will be asked to confirm this action, but you will *not* be warned if any of the fields were unique to that card. Note that you cannot delete the last remaining card of a database in this way. Again, we strongly recommend making a backup of the database before you delete a card.

23.5 Deleting an entire database

By default, Impact databases you have created will be stored within a *UserData* directory. Each database consists of a directory containing several files. To delete the database, you simply need to go into the Filer window and delete the directory with the right name. Before you do this, make sure that if you are running *Impact* the database you wish to delete is not open! Also make sure that the database is not referred to from any other database you wish to keep.

23.6 Deleting unwanted *Ovation Pro* and *Impression* merge documents

If you wish to retain a database but delete documents you have created using the **merge tool**, you will find these in the *Documents* sub-directory of the database.

23.7 Deleting unwanted action scripts

Action scripts for a database are stored within the *Actions* sub-directory of the database. The action script for a particular field has a filename corresponding to the field name.

23.8 Deleting unwanted filters, indexes and field maps

These should be deleted from within Impact by selecting them from the relevant tool, and right-clicking on the tool to open the editing window. You can then click on the *Delete* button.

Chapter 24: Advanced Impact use

24.1 System variables

The following system variables are set up in Impact's *!Run* file. Note that if you alter settings in the *!Run* file and then upgrade to a later version of Impact, your settings may be overwritten. It is recommended therefore to make a copy of your customised *!Run* file before upgrading Impact.

Impact$Path

Impact$Path defines the user's working directory for accessing databases via the program menus. This may be set to point to any combination of directories on any discs by listing them separated by commas. It should not include the *SampleData* directory unless you wish to be able to create new databases in that area. More information on how to use **Impact$Path** is given in the next section.

Impact$Write

Impact$Write may be set to define the directory in which new databases will be created. If this variable is not set, the first directory found on **Impact$Path** will be used by default, and the user will also be offered the choice of other directories from a submenu.

Impact$BackupDir

Impact$BackupDir may be set to define the directory used for storing backups of databases. The variable appears by default as the backup location setting for each newly created database, but can be overridden for each database from the backup tool window.

ImpactChoices$Path and ImpactChoices$Write

Impact will read its choices from `Choices:Impact` for preference, and if they are not found there, will look for them in `<Impact$Dir>.^.Choices` or `<Impact$Dir>.Choices`, the former also being used if there is no Choices structure on the machine. Choices are written to the standard location of `<Choices$Write>.Impact`. The following two variables can be set to override this behaviour: **ImpactChoices$Path**, a comma-separated list of locations to search to read choices, and **ImpactChoices$Write**, a directory to write choices to.

Impact$Globals

The global variables for action scripts are stored by default in the *Globals* file in the choices directory. The **Impact$Globals** variable can be set to the name of an alternative directory to use for both reading and saving.

Impact$SampleData

Impact$SampleData should point to the sample databases supplied with the system. They will appear at the bottom of the database menus, and the directory will not appear on the *Create* submenu as an option of a location for creating new databases.

LabPrint$LabelFile

LabPrint, the label printing package, will load its label definitions from the *LabPrint* directory in Choices, and write them to `<Choices$Write>.LabPrint` but this can be overridden by setting **LabPrint$LabelFile** to a filename of your choice.

24.2 Storing databases in diverse locations

By default Impact looks for user databases in a single directory called *UserData* in the same directory as the *!Impact* application. This may not be desirable in all circumstances. You may, for instance, have several distinct groups of linked databases which you wish to keep separate for clarity. You may wish to create two databases with the same name. You may wish to keep one or more databases with other documents relating to the same subject. It is easy to do all of these things by altering the **Impact$Path** system variable to contain a comma-separated list of the directories you wish Impact to look in for databases.

Foreign fields

When you have more than one directory on **Impact$Path** you need to ensure that all the databases you wish to link are contained within the same directory. Impact will not allow you to create foreign fields from a database which is not in the same directory as your local database.

Databases with the same name

You may create and open multiple databases with the same name if they are located in different directories. If they have identical card designs, the only way to tell which of the databases you are looking at is to look at the *Info* window accessed from the card menu. You should also be wary of backups. Initially the backup location for a database is set to the *BackupData* directory in the same directory as the *!Impact* application. If you do not change this default then those with the same name will get muddled up there and overwrite each other's backups. It would be advisable, therefore, to set the backup location individually for each database.

The database menu

Databases will be grouped together on the Impact database menu according to the directory in which they are held. You can choose whether all databases appear on one long menu, or whether a submenu is created for each directory on **Impact$Path**, by opting for a two-level database menu in the Choices window as described in *Chapter 21: Choices*.

Creating a new database

If there is more than one directory on **Impact$Path** then the *Create* option on the iconbar menu will offer a submenu of directories to select the one in which you wish your new database to be located.

24.3 Using Impact over a network

Although Impact does **not** permit simultaneous access of a single database by more than one user, it is possible to store databases so that they can be accessed at separate times from more than one computer. Impact will warn the user if a database is already open on another computer, allowing the user to cancel the operation. Access across a network might be done via a fileserver which users have to log in to, common in schools and workplaces, or by using peer-to-peer networking, such as *ShareFS*.

Networked fileservers requiring users to log in

This sort of system is generally found in workplaces and educational institutions, rather than at home. A network licence for Impact may be purchased, and the software can be installed on the fileserver and executed from any RISC OS computer which is connected. Impact provides the means for databases to be stored separately in the user root directory of each user, ensuring that

the databases are only available to the user who created them.

To use this mode of operation, unset the system variable **Impact$Path** in Impact's *!Run* file using the command

```
Unset Impact$Path
```

When **Impact$Path** is not set, Impact will use a directory called *ImpactWork*, located in the user root directory or URD, as defined by the fileserver at login. Within *ImpactWork* a directory called *DataFiles* will hold the databases, and a directory called *Choices* will be used for saving the user's choices.

Alternatively, you could set up **Impact$Path** (and **Impact$Write**, if required) for each user, and the choices would then use the standard choices system.

Peer to peer networks

By using ShareFS or other peer-to-peer networking, you can make the hard drives of each computer on a small network available to the others. With care, many of the facilities of an Impact database can be made available to the other machines on the network, though again, it is not possible to open the database on more than one computer at a time. How to do this is best explained by an example.

Suppose we have two RISC OS computers whose drives are shared via ShareFS. The drive of one machine is shared under the name Alpha and the other under the name Beta. (We will call the respective computers Alpha and Beta also.) Impact has been installed in `$.Impact` on Alpha and some databases have been created in `$.Impact.UserData`.

We now want to open the databases on Beta. We can see the Impact installation from Beta, as it appears in `Share::Alpha.$.Impact` but if we try running it, we will get an error saying that the copy is licensed for use on another computer. Your licence permits you to install Impact on two machines, so the first step is to dig out your Impact CD or the downloaded package and the `licence/zip` file and install on the second machine.

If you then run Impact on Beta, you will see a separate set of demo databases and will be able to create new databases on Beta's hard drive. But we want to be able to load the databases from Alpha on the copy of Impact running on Beta. To do this, we need to modify the Impact `!Run` file. Shut down Impact, and then shift-double-click on the !Impact application on the hard drive to open the application directory. Load the `!Run` file into *Edit* or your favourite text editor, and find the line which says

```
Set Impact$Path          <Impact$Dir>.^.UserData.
```

Change this to

```
Set Impact$Path          Share::Alpha.$.Impact.UserData.
```

and save the `!Run` file. When you run Impact again on Beta you will find that the databases stored on Alpha are available. New databases created on Beta will be stored on Alpha's hard drive.

There are some things, however, which will not work. You are encouraged to read the following paragraphs carefully and try some experiments before you end up setting up your databases the wrong way.

References to files will work from both machines if the files are stored on the same drive as the database. In this special case Impact stores the filename with the filing system and disc name stripped off (so starting with the `$`) and can therefore put the correct discname on the front when accessing it from either machine.

If the database is stored on Alpha, and you make a link to a file stored on Beta while running Impact on Alpha, it is likely that the file will also be accessible when running Impact on Beta,

because the ShareFS name of the disc used to access the file from Alpha will also work on Beta.

If, however, you were running Impact on Beta with the database stored on Alpha, and created a link to a file stored on Beta's hard disc, this will not open correctly when running Impact on Alpha, because Impact cannot know how Alpha refers to Beta's hard disc.

Problems also apply to the location of backups, which can be defined within the database. The simplest solution is to make sure that the Impact backup choices on Beta are set to Ignore, and perform all backup operations from Alpha instead. If you want to be more ambitious, and all of your databases store their backups in `<Impact$BackupDir>` then you can define this variable in the Impact `!Run` file on Beta to point to the appropriate directory on Alpha.

If you use global variables in action scripts you will have to decide whether you want the variables to be truly global, being the same whichever computer is used to open the databases, or whether you want the potential for the variables to take different values on the two computers. For truly global variables, you need to set the **Impact$Globals** variable in the `!Run` file on Beta to point to the Impact choices directory in Alpha instead:

```
Set Impact$Globals Share::Alpha.$.!Boot.Choices.Impact
```

It is possible to set things up in ever more complex ways, should you wish to be able to store databases on both computers. For example, you could set up **Impact$Path** on Beta so that first databases from Alpha are listed, and then databases stored locally on Beta, by using the following two lines:

```
Set Impact$Path Share::Alpha.$.Impact.UserData.
Set Impact$Path <Impact$Path>,<Impact$Dir>.^.UserData.
```

24.4 Loading databases by unconventional means

You can load a database into Impact from anywhere on your computer simply by dragging the database directory to Impact's iconbar icon. Note that copies of databases produced by the **backup tool** cannot be opened in this way because their contents are compressed.

You may wish to open a certain database every time start Impact. This may be done by including it on the command line at the very end of the *!Run* file:

```
Run <Impact$Dir>.!RunImage ADFS::HardDisc4.$.Databases.Photos
```

If you give a database name without a path, then Impact will open the first database of that name which can be found on `<Impact$Path>`:

```
Run <Impact$Dir>.!RunImage Videos
```

Alternatively, you could create an Obey file elsewhere containing a command as follows:

```
Run <Impact$Dir> ADFS::HardDisc4.$.Impact.SampleData.Videos
```

The disadvantage of this approach is that if you have several Obey files set up to open different databases, each Obey file will start another copy of Impact running and you may end up with several copies on the iconbar. This is not generally a good idea, as it wastes memory and can lead to conflicts over saving of choices, accessing files, etc.

24.5 Copying and renaming databases

Copying and renaming of databases can be done simply by copying or renaming the database directory in a Filer window when the database is closed. Even more straightforwardly, you can use the *Copy database* tool from the card menu, and this is described in *3.22 Copying databases*. There are a few traps for the unwary when renaming databases, and these are explained here.

Caution should be exercised when renaming relational databases, as if another database has

foreign fields drawn from the renamed database Impact will be unable to find those fields when opening the database.

Your database may include documents for mail merging with *Impression* or *Ovation Pro*. If so, be aware that the merge commands in the documents refer to the name of the database, and will need updating if you have renamed the database. It is fairly straightforward to do this in *Ovation Pro*, and the method is described in *10.22 Entering merge commands directly from Ovation Pro*.

You will also need to exercise care regarding backups of a database. If you wish to restore a backup using the backup tool, only backups which have the same name will be shown. Renaming the appropriate directory in your backup location at the same time as renaming the database directory avoids this problem.

If you copy a database (rather than renaming) and wish to be able to access the backups of the original version from the new one, you will also need to copy the backup directory.

If you have made a copy of a database with the same name as the original, by placing it in a different directory on **Impact$Path,** you will need to set the backup location for the copy to point to a different location from the original, otherwise the two sets of backups will get mixed up and overwrite each other.

24.6 Changing the order of texts in option, menu and browser fields

This section describes in detail how to reorder the items on a menu field. A similar method can also be applied to browser and option fields.

Suppose that on your database card you have an existing menu of *Apples, Grapes, Pears* and you wish to add *Bananas* as a fourth option. Logically you might wish your menu to remain in alphabetical order and so place *Bananas* between *Apples* and *Grapes*. Using the menu field attributes window the only way to add *Bananas* to the menu is to add it after *Pears*, in which case your menu would read *Apples, Grapes, Pears, Bananas*.

If you were to alter each menu option text so that *Grapes* changes to *Bananas*, *Pears* to *Grapes* and *Bananas* to *Pears*, you would get a menu reading *Apples, Bananas, Grapes, Pears* but you would find that all records which previously said *Grapes* now said *Bananas* and those which previously said *Pears* now said *Grapes*. This is because the menu field data for each record is actually stored as a number reflecting the position in the menu of the chosen text.

Clearly it would be nice to be able to reorder the texts in a menu without altering the data in the record! Here is a method of doing this.

1. Take a backup of your database in case anything does not go according to plan.
2. If you do not already have one on your card, create a **calculation field** of type Number with the calculation expression
   ```
   Record()
   ```
   Give the field a name. This field will display the record number of each record in the database. If you wish to retain this field for later use, you may wish to alter the number of decimal places to 0 as it looks nicer, but this is not necessary for what we are trying to do.
3. While you are in card editing mode, add the **export** and **import tools** to your toolbar if they are not already present.
4. Save the changes to the card. Create a **field map** consisting of your record number field and menu field, with the record number field being the first field in the map.
5. Make sure you have no indexes or filters active. Activate your new field map and click on the **export tool**. From the menu choose *CSV (Impact/EasiWriter)* to enable

appropriate settings. Save a CSV file to the location of your choice. This will contain only record numbers and the selected text from the menu item for each record.

6. Re-enter card editing mode and edit the texts in the menu field so that they appear in the order you require, but remain identical in wording.
7. Save the changes to the card. At this point the data in the menu field in many of your records will have been changed and will be incorrect.
8. Ensure you still have your field map selected, and click on the **import tool**. Select *Overlay* and drag your CSV file to the box marked *Drop data file here*. You should find that all your existing records are overlaid with the data from the CSV file, and the menu field in each record now contains the correct data.
9. To tidy up you can now discard the CSV file and delete the field map, and if you wish remove the export and import tools and the record number field you created.

Please note that simple filters defined on these types of field use the internal numeric representation of the data, and so these will need editing and redefining if you have reordered the texts.

24.7 Changing the type of a field

There are some circumstances in which you may wish to change the type of a field in an existing database. For instance, you may have a date field in which you need to record a date before 1900 or a number field in which you need to include letters or spaces. In order to do each of these you would need to change the type of the field to text. Or you may have an option field in which you wish to allow more than one item to be selected, so changing it to a flag field would be desirable. Or you may have an option field whose list of items is too long to be practical, which you wish to replace with a browser or menu field.

Impact does not allow you to change the type of an existing field. However, using a method similar to that outlined in the previous section, it is possible to delete the field and recreate it without losing your data.

Caution! **Do not** attempt to use this method for a field which is used in a foreign database.

1. Take a backup of your database in case anything does not go according to plan.
2. Check through your **indexes**, **filters** and **field maps** and make a note of the definitions of any which include the field whose type you wish to change, as when that field is deleted it will be removed from the index/filter/field map definition. If the field is the only one used in the definition then that index/filter/field map will be deleted entirely.
3. If you do not already have one on your card, create a **calculation field** of type Number with the calculation expression

   ```
   Record()
   ```

   Give the field a name. This field will display the record number of each record in the database. If you wish to retain this field for later use, you may wish to alter the number of decimal places to 0 as it looks nicer, but this is not necessary for what we are trying to do.
4. While you are in card editing mode, add the **export** and **import tools** to your toolbar if they are not already present.
5. Save the changes to the card. Create a **field map** consisting of your record number field and the field whose type you wish to change, with the record number field being the first field in the map.

6. Make sure you have no indexes or filters active. Activate your new field map and click on the **export tool**. From the menu choose *CSV (Impact/EasiWriter)* to enable appropriate settings. Save a CSV file to the location of your choice. This will contain only record numbers and the data from the field whose type you wish to change.
7. Re-enter card editing mode and delete the field whose type you wish to change. Create a new field of the type you desire. If this is an option, flag, menu or browser field, recreate the choice of items you need. You can give the new field the same name as one you just deleted, or a different name if you prefer. (Note that if you change the name then you will need to alter any calculation expressions which refer to the old name.)
8. Save the changes to the card.
9. Go back to the definition of your new field map, which will have lost the deleted field, and add the newly created field to the map. Ensure the field map is activated.
10. Click on the **import tool**. Select *Overlay* and drag your CSV file to the box marked *Drop data file here*. You should find that all your existing records are overlaid with the data from the CSV file, and the new field contains the data from the old one.
11. Go back through the indexes, filters and field maps which you recorded in stage 2, and recreate them or amend their definitions to include the new field, as required.
12. To tidy up you can now discard the CSV file and delete the field map you created in stage 5, and if you wish remove the export and import tools and the record number field you created.

24.8 Advanced tool editing

The tool sprites are taken from Impact's *Sprites* file. Tool sprites should have names beginning with `t_` in order to appear in the *Tools* window. There should also be a version of the tool sprite having the same name prefixed with `!t_` which shows the tool button in a style to be used when it is depressed. A selection of suitable sprites is provided, but new ones may be created by the user using *!Paint* or other graphics software.

The behaviour of each tool is defined by an **action script** which may be edited by adventurous users by selecting *Edit script* from the *Tool* sub-menu. The script name is the same as the sprite name, but without the `t_` prefix.

Note that once the tool action file is open, it may be left on screen for testing the script, when card editing has been completed. When a change has been made, save the file and click on the tool to test it, then re-edit the script if required. Note that if your text editor supports throwback, errors in the script will be shown in a throwback window, which may then be used to indicate and display the line containing the error.

Note that any alterations to tools will affect **all** databases which use those tools, not just the one from which you edited the scripts. Also, if you upgrade to a later version of Impact you will need to copy your scripts/sprites elsewhere before upgrading to avoid their being overwritten.

Appendix 1: Field types

The field types available within an Impact database are listed below, together with the disc space used, the maximum numerical range, etc.

Field type	Disc space	Range	Maximum length
Text	user defined		up to 999 characters
External file	user defined		up to 255 characters
Internal file	variable		
Actions	none		
Calculation	none		
Number	8 bytes	±1.79E308	up to 49 digits
Date	5 bytes	1900-2247	
Flag	4 bytes	0-31	
Option	4 bytes	0-31	
Menu	4 bytes	0-31	
Browser	4 bytes	0-31	
Sequence	12 bytes		up to 12 characters

The disc space shown above is the space taken *per record*, and applies whether any data is stored in a record or not. In the case of images, additional space is also used for all images stored.

Appendix 2: Mouse and key actions in record cards

The table below shows the effect of various key presses, drags and mouse clicks on fields in Impact cards, and the differences arising having partial text selection enabled for text fields.

Action	**Text field with text selection enabled**	**Text selection is disabled, or the field is *not* a text field**
Key presses		
Ctrl-A	Select whole field	—
Ctrl-C	Copy selected text	Copy whole field (if text field)
Ctrl-U	Clear field, for a single-line icon	Clear field
Ctrl-V	Paste at caret	Paste at caret (if text field)
Ctrl-X	Cut selected text	Cut whole field (if text field)
Ctrl-Z	Clear selection	—
F10	Delete current line	Delete current line (if text field)
Dragging within an icon		
Drag	Select text, or move selected text to new caret position. *Note that while another card design is being edited, selection is temporarily disabled in all cards.*	Start export of field contents
Shift-drag selected text	Copy selected text to new caret position	—
Dragging from an icon		
Drag	Export selected text	Export whole field
Alt-drag	Export whole field	Export whole field
Shift-drag	Export selected text and delete	Export whole field and clear field
Dragging to an icon		
Drag or ***Shift-drag***)	Insert text at pointer	Replace whole field
Ctrl-drag or ***Shift-Ctrl-drag***	Insert text at pointer	Append to field (if text field); Replace whole field (otherwise)
Mouse clicks with Select		
Double-click	Select word (keep clicking to extend the selection)	Open find tool
Alt-double-click	Open find tool	Open find tool
Ctrl-click	Place caret	If a foreign key field, open related database card for that record
Ctrl-double-click	Edit field in external editor (OLE)	—

Appendix 3: Date format codes

The full list of date format codes is as follows. Upper or lower case may be used.

`%24`	The current hour in 24 hour clock format.
`%12`	The current hour in 12 hour clock format.
`%MI`	The number of minutes past the hour.
`%SE`	The number of seconds past the minute.
`%CS`	The number of centiseconds (hundredths of a second).
`%AM`	The letters ‘am’ or ‘pm’ as appropriate.
`%PM`	The letters ‘am’ or ‘pm’ as appropriate.
`%WE`	The day of the week string, in full, e.g. Tuesday.
`%W3`	The first three characters of the weekday name.
`%WN`	The weekday’s number, Sunday = 01.
`%DY`	The day of the month, e.g. 01.
`%ST`	The suffix ‘st’, ‘nd’, ‘rd’, or ‘th’ as appropriate.
`%MO`	The month name string in full, e.g. August.
`%M3`	The first three characters of the month name.
`%MN`	The month number, January = 01.
`%CE`	The century, e.g. 19.
`%YR`	The year within the century, e.g. 89.
`%WK`	The week number (Mon-Sun) in the year.
`%DN`	The day number in the year, e.g. 048.

In addition, a ‘z’ may be included before relevant fields to suppress any leading zeros, as in `%zdy`

Note that though `%WK` and `%DN` may be used for display purposes, Impact will not accept dates input in these formats.

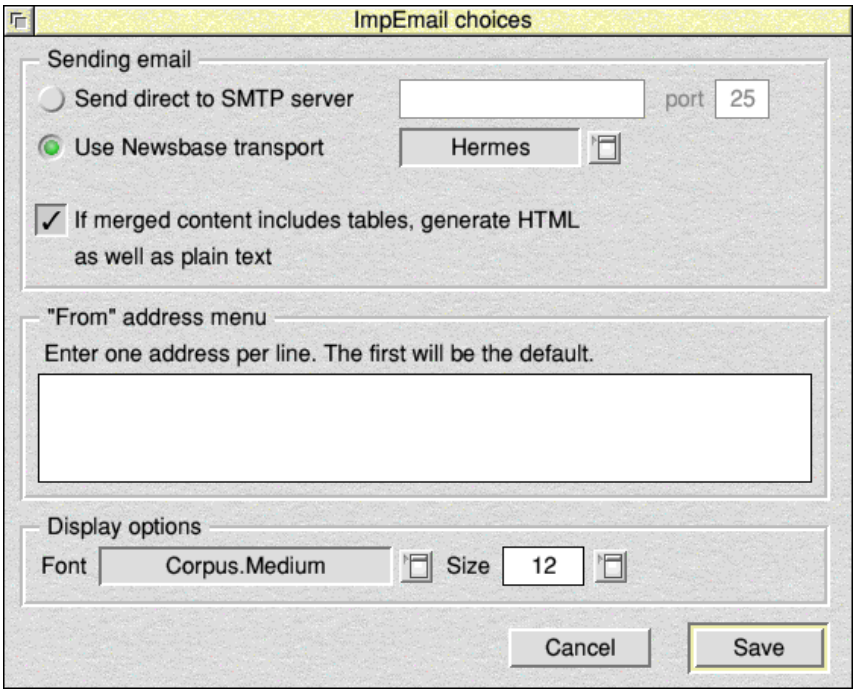
The formats `%M3` and `%W3` give abbreviations which are three letters long when the operating system is set to use English, but in other territories the abbreviations may be of differing lengths.

Appendix 4: *ImpEmail* manual

ImpEmail is a simple application for composing and sending e-mails, designed for mail-merging in conjunction with *Impact,* or any application which supports the *Impulse* protocol such as *Powerbase*.

A4.1 Configuring ImpEmail

The first time you run ImpEmail it will prompt you to set up your choices, to determine your method of sending messages, and via which SMTP server or e-mail transport application your messages will be sent.



There are two options for how e-mail is sent off. One is for ImpEmail to communicate directly with your ISP's SMTP server; the other is to use a Newsbase transport such as *POPstar* or *Hermes*. If you choose to use a Newsbase transport, messages will be placed in that transport's queue for sending later; if you use direct SMTP, messages will be sent immediately if the server can be contacted.

If you choose to use direct SMTP, enter the name of your SMTP server (e.g. smtp.myisp.co.uk) and change the port number if necessary. Note that ImpEmail does not support SMTP servers which require authentication. If your server requires a username and password to be sent, we recommend you use *Hermes* instead of direct SMTP. *Hermes* is part of the *NetFetch* application, which may be purchased from R-Comp.

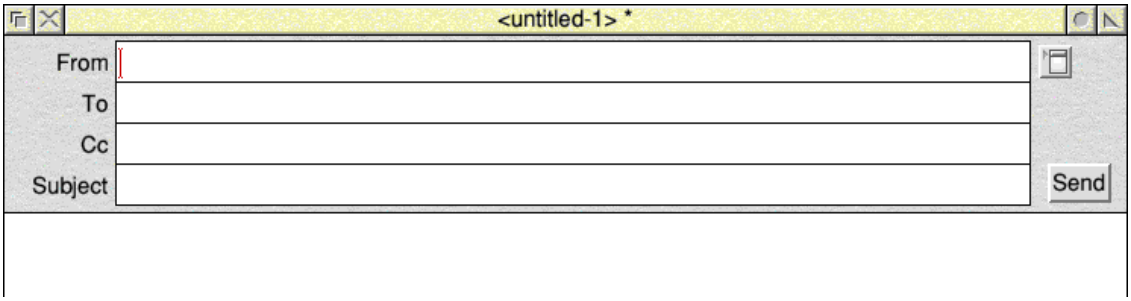
If you prefer to use a Newsbase transport, you may have to run your e-mail software first to ensure ImpEmail can find all the possible transports. All you then have to do is select the one you use from the menu if that has not already been placed in the box. You do not usually need to be running any other software while using ImpEmail: this is just to enable the initial configuration.

To change the configuration at a later date, simply choose *Choices...* from ImpEmail's iconbar menu.

A4.2 Composing messages

To create a new message click on the ImpEmail iconbar icon. An editor window will open

showing *From*, *To, Cc* and *Subject* fields, with a large box for the message.



Format of From, To and Cc fields

You may simply put the e-mail address of an individual in any of these fields:

```
support@sinenomine.co.uk
```

If you wish to include multiple addresses in a To or Cc field, separate the addresses with comma and space.

It may help you to identify the recipients if you also include their names. Each name should be placed inside "" and then the e-mail address should follow it, enclosed in <>. For example:

```
"Joe Bloggs" <spamhater@address.co.uk>, "Arthur C. Clarke" <
author@2001.com>
```

We will explain the format of **merge commands** later in this manual, but for reference, if you are inserting a name and e-mail address from the Impact *Addresses* database using a merge command, the To field might read

```
"{merge ":Impact.Addresses Fetch Name"}" <{merge
":Impact.Addresses Fetch email"}>
```

You can see an example of inserting both name and e-mail address in the *Addresses* sample database supplied with Impact. Look at the merge document called VideoList.

With the database *Powerbase* the merge commands are slightly different. This example assumes the e-mail address is in a field called EMAIL and the name is in a field called NAME:

```
"{merge ":Powerbase GetField NAME"}" <{merge ":Powerbase
GetField EMAIL"}>
```

Setting up a default From address

In the Choices window you can enter one or more addresses to be available on a menu for easy insertion in the From field. The first address will be the default value, and others can be chosen by clicking on the menu button next to the From field. If you want a menu of addresses but prefer the default to be blank, simply enter a blank line at the top of your list of addresses. You may wish to store your name as well as your e-mail address in the Choices list, e.g.

```
"John Smith" <johnsmith@myhomeisp.co.uk>
"J.R. Smith" <jr.smith@work-address.com>
```

Display options

You may change the current font name and size by clicking ***Menu*** over the editor window. The default font to be used can be set in the Choices window. Note, as the messages are sent in plain text this is just the font that you, the editor, see, and it is not transmitted to the recipient of the message.

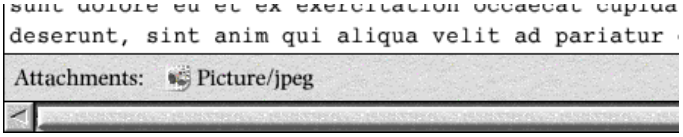
Shortcut keys

Ctrl-A	Select whole text
Ctrl-C	Copy selected text
Ctrl-V	Paste at caret
Ctrl-X	Cut selected text
Ctrl-Z	Clear selection
Ctrl-F2	Close the editing window
F3	Save: display the save dialogue
Ctrl-F3	Save: saves straight away to the location the file was loaded from
F10	Delete current line

ImpEmail supports the global clipboard using the shortcut keys shown above, and also the drag-and-drop protocol. Dragging a selected region to another part of the same e-mail document will move it. Dragging with ***Shift*** held down will copy the region. When dragging a selected region and dropping it outside the e-mail document (for example into another e-mail, or to another application) the operations are reversed: a plain drag will copy the selected text while a Shift-drag will move it. This standard is supported by many other RISC OS applications, prominent examples being EasiWriter and TechWriter.

A4.3 Attachments

You can drop files from a Filer window in the attachment area at the bottom of the message editing window. Each file will appear with the appropriate icon and the filename alongside:



A menu over the attachments area allows you to select and remove attachments, and edit their properties. You may also edit the properties of the attachment by double-clicking the name or icon.

How and when ImpEmail attaches files

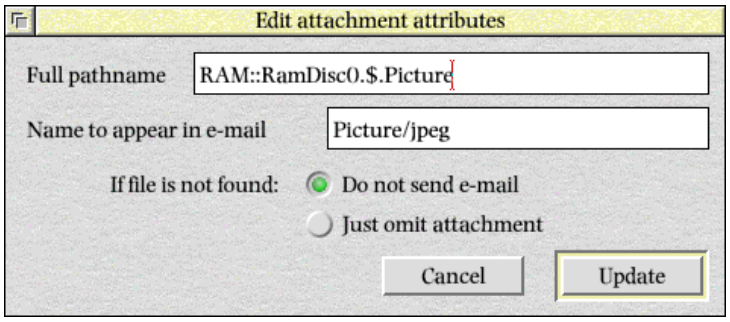
It is important to realise that ImpEmail only actually stores the path and filename of each attachment. The complete content of an attached file is only placed into the e-mail at the point you send it. One consequence of this is that you cannot drag files direct from an application and attach them to an e-mail: they may only be dragged direct from the Filer.

It also means, however, that if the content of any attached file changes before the message is sent, it is the version of the file at the point the message is sent which will be attached. This could be useful if you have regular communications which you need to send to people. For example, if you are running a club, you might have a welcome e-mail message for new members, to which the club's membership guide is attached as a PDF. In your membership database, you could store the welcome message as an ImpEmail document, with the attachment pointing to the location on your hard drive where the guide may be found. If you update the guide in the mean time, the next time you send a welcome message to a new member the latest guide will be attached.

Editing attachment attributes

From the menu over an attachment there is an Edit option. This opens a window where the

attributes of the attachment can be altered. You can also open the window by a double-click on the file or icon.



The full path and file name are shown in the first box in the form. This is used to locate the file when ImpEmail needs to attach the content when building the message for sending.

The next box shows the name by which the attached file will appear to recipients of the e-mail. This is usually just the name of the file without the details of the disc name or directory names. You can put whatever you like in this box. Note that when you drop a file in the attachment area, ImpEmail may adjust the name as it will appear to recipients to help ensure that they are able to use the file correctly. The file type of an attachment is defined in an e-mail using the Content-Type header, in a cross-platform standard way. E-mail software widely used on Windows PCs ignores this and instead relies on the filename extension, so ImpEmail will apply the appropriate extension for you when you attach the file.

In the screenshot above you can see that while the RISC OS filename of the file is simply "Picture", ImpEmail has added "/jpeg" to identify the file type to users of Windows e-mail software. (The slash will be converted to a full-stop on sending.)

You may also define what will happen when the e-mail is sent if the attached file cannot be found or read at that stage. To continue the previous example, if you move the membership guide document, forgetting that it is linked from your new member's welcome message, it would be best for ImpEmail to alert you to this by refusing to send the message and giving an error.

There may be circumstances, for example if the attachment is drawn from a file field in the database, where you would wish to send the e-mail message anyway but omitting the attachment. You can choose what will happen using the radio buttons at the bottom of the attachment attributes window.

A4.4 Merge commands

The whole reason for the development of ImpEmail is to support Impulse-based mail merging from database applications such as *Impact* and *Powerbase*. This section explains how to format them and gives examples.

General format of merge commands

Both Powerbase and Impact allow you to drag-and-drop fields and other merge data into the ImpEmail editing window. Merge commands can appear in any area of the ImpEmail document, including the From, To, Cc and Subject fields.

In ImpEmail the merge commands will be shown as

```
{merge ""}
```

with the merge command appearing between the double-quotes. You can edit the commands directly in ImpEmail if you wish. Double-quotes included in merge commands must be doubled up to be interpreted correctly.

As merge commands are enclosed in braces, if you wish to include an opening brace in your message for other purposes you will need to precede it by a backslash:

```
\{
```

to ensure that ImpEmail does not try to interpret it as a merge command. Similarly if you wish to include a backslash in your message it must be preceded by another:

```
\\
```

Merging data from Impact

To include fields or tables from Impact in the message, use Impact's **merge tool** to create merge commands (see *Chapter 10: Mail merging and formatted reports*). Alternatively, you can type commands direct into ImpEmail, following the guidance in that chapter as to the syntax of Impact's merge commands.

Tables and HTML e-mails

As well as including individual fields and calculation expressions, Impact supports merging of entire tables of data with a single command. See the VideoList example in the Addresses sample database supplied with Impact, which lists the videos on loan to an individual.

Displaying the tables accurately in a plain text e-mail requires the recipient to view the message using a fixed-width font. By default ImpEmail is configured to use the Acorn font Corpus, which is very similar to Courier on other operating systems, but that will only show you what the message will look like in ideal conditions: the recipient may be using a variable-width font. ImpEmail makes a good job of displaying tables as plain text, including handling multi-line text fields and correctly aligning columns of numbers, but if there are too many columns to display the resulting message will inevitably be hard to follow.

Most e-mail software on other platforms can display HTML (the mark-up language which is used for building web pages). If ImpEmail detects that a table has been merged into the message, then as well as building a plain text version of the e-mail, an HTML version will be included, which will tend to display better. You can turn this off via the Choices window.

Table merging into the To and Cc fields

ImpEmail also supports table content being merged into the fields defining the e-mail addresses of the recipients of the message. The table content will be scanned automatically to identify which column contains the e-mail address. Any other columns will be joined to form the person's name. This feature allows you to send a single copy of a message to several people at once, with them all being able to see and reply to the other recipients.

Be careful to ensure that all the rows of the table contain valid e-mail addresses. If any are syntactically invalid or empty, then the whole message will be rejected before transmission. If all the e-mail addresses basically look right, but one or more happen not to exist, then it will depend on the transmission method used as to whether the message is delivered successfully to the other recipients.

Merging attachments

You are not limited to attaching a particular static file to ImpEmail messages by dragging it and dropping it in the attachments area of the editor window. If you open the menu over the attachments area, there is a "New attachment..." option. ImpEmail simply records the path to the file you wish to attach, and the boxes in the form can accept merge commands just like the other areas of the ImpEmail editor. So if your database includes a field containing a filename, you can just merge this field into the "Full pathname" box to have the file attached to the

message during the merging process. See *Section 10.24 Merging and attachments in ImpEmail* of the Impact manual for a full description of this process.

Merging data from Powerbase

The merging process is described in chapter 9.3 of the Powerbase reference manual. That chapter currently only refers to *Impression* and *Ovation Pro* but the same principles apply to ImpEmail.

To create a simple merge command to fetch a field, shift-drag the field from the Powerbase record card and drop it in ImpEmail's document window. The merge command will look something like this:

```
{merge ":Powerbase GetField FIELDNAME"}
```

Note that, unlike with Impact, Powerbase has no way of specifying which database should be used for the fetching of the fields. You must make sure you have the correct database loaded in Powerbase to make sense of the merge commands in a saved ImpEmail document.

Powerbase has a few other merge command variants which can be accessed via Powerbase's main menu using the option 'Insert merge tags...'. These allow you to fetch data from validation tables and to move on to another database record during the merge process.

Examples:

```
{merge ":Powerbase PreviousMatch"}
{merge ":Powerbase NextMatch"}
{merge ":Powerbase GetExpanded COMP,4"}
```

A4.5 Saving messages

Messages can be saved by pressing ***F3*** or choosing *Save* from the menu over the editor window. Pressing ***Ctrl-F3*** will save back to the location the file was loaded from. To load a saved message back into ImpEmail just double-click on it.

Messages do not have to be complete to be useful for mail merging. You could, for example, set up your From address and an e-mail signature, and save the message to *Impact's* choices window to make a default message which would be used when creating a new merge document.

You can also save messages after they have been sent, complete with the merged values from the database, by accessing ImpEmail's status window. See below for further details. The messages are saved in a format suitable for loading into your standard e-mail software.

A4.6 Sending messages

To send a message exactly as it is currently shown in the ImpEmail window, click on the *Send* button. If you need merge commands to be processed first the options depend on the application.

Impact

There are three ways of processing the merge commands:

1. Use the *Test* button in Impact's **merge tool**. The commands will be processed just for the current record of the database. You can check over the message and then click ImpEmail's *Send* button.
2. Use the *Test* button and then the *Fix* button from the **merge tool**. The commands will be processed for the current record, but the message is returned to an editable state, so you can make any further adjustments before clicking *Send*. Note that

saving the message in this state would overwrite the version containing the merge commands: be careful! You do not have to save in order to send.

3. Use the *E-mail* button in the **merge tool**. The merge commands will be processed for the current record, for all filtered records, or for the selected records, depending on the settings in the merge tool window. The e-mail or e-mails will be sent to the outgoing mail queue without further intervention. This is the method you would usually use for sending messages in bulk.

Powerbase

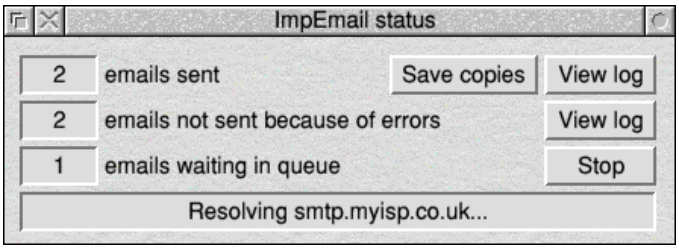
The Data merge window is opened by dropping a saved ImpEmail document onto the Powerbase record card. There are several different ways to process the merge commands, depending on what you want to achieve.

1. Clicking the *Merge* button will process the merge commands for a single record. You can then send the message by clicking Powerbase's *Print* button or ImpEmail's *Send* button. Using Powerbase's tool pane you can move to another record and the merge commands will be reprocessed. In this way you can send a hand-picked selection of e-mails.
2. Clicking the *Merge* button and then clicking *Edit* will process the commands for the current record but the message is returned to an editable state, so you can make any further adjustments before clicking ImpEmail's *Send* button. Note that saving the message in this state would overwrite the version containing the merge commands: be careful! You do not have to save in order to send.
3. To send an e-mail for each record in the database, ensure the *Print all* option is selected before clicking *Merge*.

Message transmission and logging

If you are using a Newsbase transport ImpEmail will put the messages in the queue to be sent by *POPstar* or *Hermes* when you next send a batch of e-mails off. You can view the queue from *Messenger Pro* and edit or remove the messages while they are waiting to be sent.

If you are using direct SMTP, ImpEmail will attempt to send the messages as soon as possible. If ImpEmail is unable to communicate with your SMTP server, the messages will remain in ImpEmail's queue and you can send them later or next time you run ImpEmail. A status window will appear whenever a message is sent by SMTP, so that you can monitor the progress of the transmission. If you need to stop the transmission, click on *Stop*. This button will read *Send* if there are messages waiting in the queue after an internet communication problem.



ImpEmail keeps a record of the messages sent, with a log for successful messages and a log for unsuccessful messages which details the reasons for failure. The logs are cleared out the next time ImpEmail is run. After sending an e-mail or a batch of e-mails using *Impact's* merge tool, ImpEmail's status window will open to show a count of successes and failures, and to allow you to view the logs. If you need to access the status window at other times it can be opened from ImpEmail's iconbar menu.

You can also save the successful e-mails by clicking on *Save copies*, which provides you with a file of type &B86 which can be loaded into *Messenger Pro*, *Pluto* or other e-mail software, or saved to disc. You can keep the file separately and double-click to view the messages using *Messenger Pro*, or the file may be dropped into an e-mail folder in *Messenger Pro* to add the e-mails to that folder. To import the messages into a *Pluto* mail folder just drop the file on Pluto's iconbar icon. The format of the file is very simple, so if you have no other way to read it using a text editor is quite sufficient.

Appendix 5: Obtaining support

E-mailing Sine Nomine Software

You can e-mail us direct by clicking the button labelled ‘Send e-mail’ which you will find on the Info window, which you can reach from the application’s iconbar menu. We will try to reply promptly, but there are occasions when we are busy or on holiday.

Discussion list

An e-mail list, hosted by Yahoo, has been set up to support Impact and ImpEmail. It is not a very busy list, but has quite a few users as members, so we would encourage you to join.

Please use this list to ask any questions you may have regarding Impact, ImpEmail or LabPrint. We monitor the list and will respond when needed. All users can benefit from the discussions and ideas which result.

To subscribe to the e-mail list, simply go to the Impact List home page at Yahoo Groups, and click on the ‘Join This Group!’ button. This method requires a Yahoo username. If you do not have one and do not wish to create one, you can join the mailing list by sending a message to the group's subscribe address, impact-list-subscribe@yahoogroups.com.

Index

* commands 134

A

Abort action command 143
Abs function 120
ACos function 121
Action buttons 17, 20, 90
  attributes 99
  editing legends 91
  menu-style 90, 105
  using as functions 146
Action commands 132
Action scripts
  commands *see* Action commands
  deleting 182
  from flag fields 97–98
  from option fields 97–98
AddMonths function 119
AddSlashes function 115
AddYears function 119
Age function 118
AND 36
ASCII function 117
ASin function 121
AskDate function 112
AskNum function 112
AskText function 112
ATan function 120
Attachments 195
  mail merging 71
Automatic backups 179

B

Backup of data 25, 29, 178
  choices 175
  over a network 186
  restoring data 180
BaseConvert function 122
Beep action command 143
Boxes on record cards 90
  attributes 100
Break action command 143
Browser fields 89
  entering data 45
  importing 169
  in calculations 130
  reordering texts 187
Browsing records 14, 28
Button function 126

C

Calc merge command 66
Calculation fields 90
  attributes 101
Calculation filters 34
  in merge commands 64
Calculation merge command *see* Calc merge command
Calculations 107
Caller display 171
  choices 176
Card attributes window 84
Card function 127
Card menu 27
CardFile function 124
Cards *see* Record cards
CardText function 113
Caret order *see* Fields (caret order)
CField function 127
Char function 117
Choices 173
Clear action command 141
Clearing fields *see* Fields (clearing)
Close action command 141
Close tool 28
Closing a database *see* Databases (closing)
Comments in action scripts 135
Compound filters 34
Concatenating fields

field maps 40
Conditional calculations 129
Conditional commands 140
Constants 107
Context *see* Execution context
Control codes in text strings 107
Copy action command 141
Copy tool 28
Copying a database 31
see also Databases (copying and renaming)
Copying fields
card editing 85
Cos function 120
Count function 111
Creating a database *see* Databases (creating)
CSV export 154
CSV import 159

D

Data
editing *see* Editing a record
entering new data *see* Entry tool
saving *see* Saving data
Data entry tool *see* Entry tool
Databases
choices on opening 174
closing 25, 31
copying and renaming 31, 186
creating 20, 82
database menu 176
deleting 182
loading by unconventional means 186
opening 13
relational *see* Relational databases
storing in diverse locations 184
DataName function 113
DataPath function 113
Date fields 87
attributes 95
entering data 43
format codes 192
importing 169
Date format codes 192
Date function 118
Date merge command 67
Day function 118
DaysInMonth function 118
Dbox action command 140
Degs function 121
Delete action command 137
Delete record tool 28
Deleting a database *see* Databases (deleting)
Deletion 181
Dependent fields 103
DFormat function 119
Dialogue boxes
displaying from action commands 140
Discussion list 201
Display action command 137
DocClear action command 139
DocFix action command 138
DocMerge action command 138
DocPrint action command 138
Drag and drop 25, 42, 154

E

E-mail 147, 193
EasiWriter
exporting data to 156
Editing a record 31, 42
Else action command 140
Entry action command 141
Entry tool 14, 28, 42
Equal to 36
Error action command 143
Excel
exporting data to 156
Execution context 132
merge commands 60
Exists function 128

Exp function 121
Export action command 143
Export formats 155
Exporting data 24, 29, 154
Expressions 107, 109
  action commands 132

F

Fact function 120
Fetch merge command 61
Field editing menu 91
Field function 127
Field maps 16, 24, 29, 40
  creating and modifying 40
  deleting 41
  effect on importing 167
  selection on opening a database 174
Field types 23, 87
  changing 188
  technical data 190
Fields
  assigning values in action commands 132
  attributes 91
  caret order 93, 104
  clearing 45, 181
  default entries 93
  editing legends 91
  help text 91
  justification 93
  moving 92
  names 92
  order on menus 175
  placement grid 176
  reading values in calculations 108
  removing from a card 181
Fields action command 135
Fields function 127
FieldText function 113
FieldValue function 124
File fields 87
  attaching to e-mails 71
  attributes 96
  clearing data 45
  editing legends 91
  entering data 43
  in calculations 130
  over a network 185
File fields, internal
  importing 169
FileContent function 125
FileName function 124
FileSize function 125
FileType function 125
Filter action command 135
Filter function 127
Filter tests 35
Filters 15, 23, 29, 33
  creating and editing 33
  deleting 35
  effect on importing 168
  flag fields 34
  hiding from menus 35
  in merge commands 61, 64–65
  selection on opening a database 174
  use in indexes 38
Filters function 127
Find action command 136
Find tool 16, 30
Fireworkz
  exporting to 158
First action command 136
Flag fields 89
  attributes 97
  editing legends 91
  entering data 45
  filtering 34
  importing 169
  in calculations 129
Flag function 126
  use in filtering flag fields 34

Foreign fields 103
attributes 104
clearing 106
creating 103
entering data 104
importing 169
Format function 116
Fract function 120
Functions 110

G

Global action command 142
Global variables 142
Greater than 36

H

Help while using Impact 13
Hot keys *see* Keyboard shortcuts
HourGlass action command 144
Hrs function 118
Hyp function 121

I

Iconbar menu 13
If action command 140
Image fields 88
attributes 96
clearing data 45
entering data 44
Impact$BackupDir 179, 183, 186
Impact$Globals 183, 186
Impact$Path 183–185
Impact$SampleData 183
Impact$Write 183
ImpactChoices$Path 183
ImpactChoices$Write 183
ImpEmail 193
mail merging example 55
merging attachments 71
Importing data 24, 29, 159
Impression reports *see* Merge tool
Index action command 135
Index function 126
Indexes 15, 24, 29, 37
creating and modifying 37
deleting 39
effect on importing 168
in merge commands 61, 64–65
selection indexes 38
selection on opening a database 174
Indexes function 127
Insert function 115
Int function 120
IsNull function 128

K

Key fields 103
Keyboard shortcuts 26
KeyDown function 126
KillQueue action command 140

L

Label design *see* Labels (designing)
Label tool 29
LaBella 156
Labels 19, 24, 75
action commands 139
default label style 77
designing 77
exporting to LaBella 156
field maps 76
printing 76
queuing 77
saving 76
using images 79
using the merge tool 73
Labels on record cards 82, 90
attributes 100
LabPrint action command 139
LabPrint application 75
LabPrint$LabelFile 183
LabQueue action command 139

LabStyle action command 139
Last action command 136
LastButton function 126
LCase function 114
Len function 113
Less than 36
Local action command 142
Local variables 142
Log function 121
Log10 function 121
LTrim function 115

M

Macro action command 139
Macro function 128
Mail merging 19
  see also Merge tool
Maths functions 119
Max function 122
Menu fields 89
  attributes 98
  entering data 45
  importing 169
  in calculations 130
  reordering texts 187
Menu function 112
Merge commands
  dragging from Impact 61
  entering from Impact 60
  entering from ImpEmail 71
  entering from Impression 68
  entering from Ovation Pro 69
Merge documents
  deleting 182
  templates 177
Merge tool 18, 24, 28
  permanent merging 58
  previewing documents 58
  printing documents 59
  selecting data records 59
  selecting documents 57
MergeTo action command 138
Message action command 144
Min function 122
Mins function 117
Modem action command 144
Modified function 123
Money function 117
Month function 118
Multi-line fields *see* Text fields (multi-line)
MySQL
  exporting data to 157

N

Networked use of Impact 184
Next action command 136
Next function 123
NOR 36
Not equal to 36
Notes fields *see* File fields
Null values 110
Number fields 87
  attributes 94
  entering data 43

O

Operators 130
Option fields 89
  attributes 98
  editing legends 91
  entering data 45
  importing 169
  in calculations 130
  reordering texts 187
OR 36
Ovation Pro reports *see* Merge tool
Overlay 167

P

Partial match 36
Passwords 22, 84

Pause action command 144
Peer to peer networks 185
Phone function 129
Pi function 122
Pow function 122
Previous action command 136
Previous function 123
Printer action command 138
Printing labels *see* Labels (printing)

Q

Query function 112
Quick search tool *see* Find tool

R

Rads function 121
Range 36
Record action command 136
Record cards 22
  card attributes window 84
  creating 82
  deleting 181
  editing the design 83
  multiple cards 84
Record function 122
Record merge command 63
Records
  deleting 181
  selection on opening a database 174
Relational databases 103
  displaying related data 17
  example 14
  finding related data 30
  making relational links 17
Renaming databases *see* Databases (copying and renaming)
Replace function 115
Replacing data *see* Search and replace
Reports using Impression or Ovation Pro *see* Merge tool
Restoring data 180
  from corrupt databases 180
Return action command 144
Return key 173
RomanNumber function 116
Round function 119
RTrim function 115

S

Safe function 123
Save action command 137
Save tool 28
Saving data 14, 45
  updating records automatically 173
  using Return key 173
Scan action command 137
Scan merge command 66
Search and replace 46
Search tool 28, 46
  see also Find tool
Searching
  complex searches *see* Search and replace
  single fields *see* Find tool
Secs function 117
Select action command 142
Select tool 28
Sequence fields 90
  attributes 100
SerialDev action command 144
SerialSend action command 145
Setting a password *see* Passwords
Shortcuts *see* Keyboard shortcuts
Show action command 141
Simple filters 33
Sin function 120
Sprites *see* Image fields
Sqrt function 121
Star commands 134
String functions 113
StripSlashes function 116
SubText function 114

Syntax
action commands 132
calculations 107
functions 110
merge commands 61
Sys function 117
System variables 183

T

Table action command 145
Table functions 111
Table merge command 64
Table tool 29
Tables 16, 23, 29, 49
choices 176
of search results 48
Tan function 120
TechWriter
exporting data to 156
TempFields action command 135
TempFilter action command 135
Text fields 87
attributes 93
entering data 43
multi-line 83, 94, 156, 158
placing 82
Text function 114
Text selection 177
TextDay function 119
TextMonth function 119
Time fields 87
attributes 95
format codes 192
Today function 117
Tool action command 145
Toolbar 14, 22, 27
adding or removing 86
editing 86
Tools 27
help text 86
tool action scripts 189
tool sprites 189
ToolState function 126
Total function 111
TotMax function 111
TotMean function 111
TotMin function 111
TotSqr function 111
TotStd function 111
Trace action command 145
Trim function 115
Trunc function 119
TSV export 154
TSV import 159

U

UCase function 114
Url action command 145
UrlEncode function 116

V

Val function 116
Valid function 128
Variables 110
creating 142

W

Web sites
launching in a browser 145
While action command 140
WimpPoll action command 145
Word function 114
Writable fields
entering data 43
making non-writable 93

X

XOR 36

Y

Year function 118